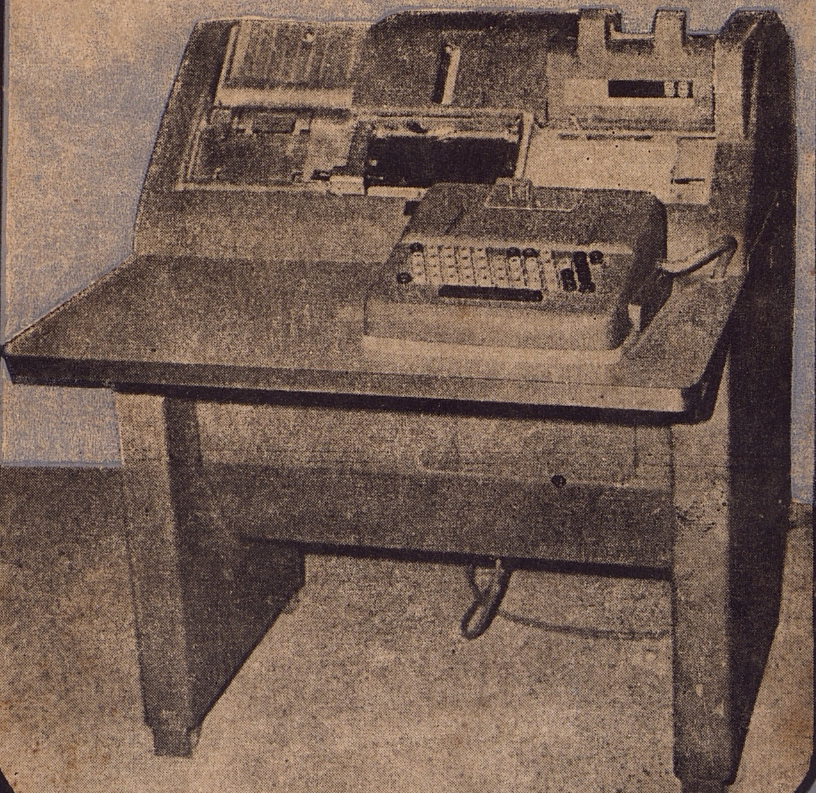
கம்ப்யூட்டரும் அதன் பயன்களும்

(COMPUTERS AND THEIR USES)

க. விசுவநாதன்



தமிழ்நாட்டுப் பாடநூல் நிறுவனம்

த. பா. நி. (க. வெ.) வரிசை எண் 625

கம்ப்யூட்டரும் அதன் பயன்களும்

(திருத்தப்பட்ட பாடத் திட்டத்தின்படி வெளியிடப்படுகிறது)

ஆசிரியர்
க. விசுவநாதன், எம்.ஏ.,எம்.எஸ்ஸி.,
சிஸ்டம்ஸ் புரோகிராமர், கம்ப்யூட்டர் சென்டர்,
பொறியியற் கல்லூரி, கிண்டி,
சென்னை.

தமிழ்நாட்டுப் பாடநூல் நிறுவனம்

First Edition — May 1975

T.N.T.B.S. (C.P.) No. 625

© TAMILNADU TEXTBOOK SOCIETY

COMPUTORS AND THEIR USES

K. VISWANATHAN

Price Rs. 8-60

Published by the Tamilnadu Textbook Society under the Centrally Sponsored Scheme of Production of books and literature in regional languages at the University level of the Government of India in the Ministry of Education and Social Welfare (Department of Culture), New Delhi.

Printed by :

Sri Surabhi Printers,

39/12, Lake View Road, Madras-33.

அணிந்துரை

திரு. இரா. நெடுஞ்செழியன்

(தமிழகக் கல்வி அமைச்சர்)

தமிழைக் கல்லூரிக் கல்வி மொழியாக ஆக்க[illegible] தாண்டுகள் ஆகிவிட்டன. குறிப்பிட்ட சில கல்லூரிகளில் பட்டப் படிப்பு வகுப்புவரை மாணவர்கள் தங்கள் பாடங்கள் அனைத்தையும் தமிழிலேயே கற்றுவருகின்றனர். 1969ஆம் ஆண்டிலிருந்து அறிவியல் பாடங்களையும் தமிழிலேயே கற்பிக்க ஏற்பாடு செய்துள்ளோம். தமிழிலேயே கற்பிப்போம் என முன் வந்துள்ள கல்லூரி ஆசிரியர்களின் ஊக்கம், பிற பல துறை களில் தொண்டு செய்வோர் இதற்கெனத் தந்த உழைப்பு, தங்கள் சிறப்புத் துறைகளில் நூல்கள் எழுதித்தர முன்வந்த நூலாசிரியர்கள் தொண்டுணர்ச்சி இவற்றின் காரணமாக இத் திட்டம் நம்மிடையே மகிழ்ச்சியும் மன நிறைவும் தரத்தக்க வகையில் நடைபெற்று வருகிறது. இவ்வகையில், கல்லூரிப் பேராசிரியர்கள் கலை, அறிவியல் பாடங்களை மாணவர்களுக்குத் தமிழிலேயே பயிற்றுவிப்பதற்குத் தேவையான பயிற்சியைப் பெறுவதற்கு மதுரைப் பல்கலைக் கழகமும், சென்னைப் பல்கலைக் கழகமும் ஆண்டுதோறும் எடுத்துவரும் பெருமுயற்சியைக் குறிப் பிட்டுச் சொல்லவேண்டும்.

வரலாறு, அரசியல், உளவியல், பொருளாதாரம், மெய்ப் பொருளியல், புவியியல், புவியமைப்பியல், மனையியல், கணிதம், இயற்பியல், வேதியியல், உயிரியல், வானியல், புள்ளியியல், விலங்கியல், தாவரவியல், பொறியியல், சட்டம் ஆகிய எல்லாத் துறைகளிலும் மூல நூல்கள் மொழிபெயர்ப்பு நூல்கள் என்ற இரு வகையிலும் தமிழ்நாட்டுப் பாடநூல் நிறுவனம் வெளியிட்டு வருகிறது.

இவற்றுள் ஒன்றான '**கம்ப்யூட்டரும் அதன் பயன்களும்**' என்ற இந் நூல் தமிழ்நாட்டுப் பாடநூல் நிறுவனத்தின் 625 ஆவது வெளியீடாகும். கல்லூரித் தமிழ்க் குழுவின் சார்பில் வெளியான 35 நூல்களையும் சேர்த்து இதுவரை 660 நூல்கள் வெளிவந்துள்ளன. இந் நூல் மைய அரசு கல்வி, சமூக நல அமைச்சகத்தின் மாநில மொழியில் பல்கலைக் கழக நூல்கள் வெளியிடும் திட்டத்தின்கீழ் வெளியிடப்படுகிறது.

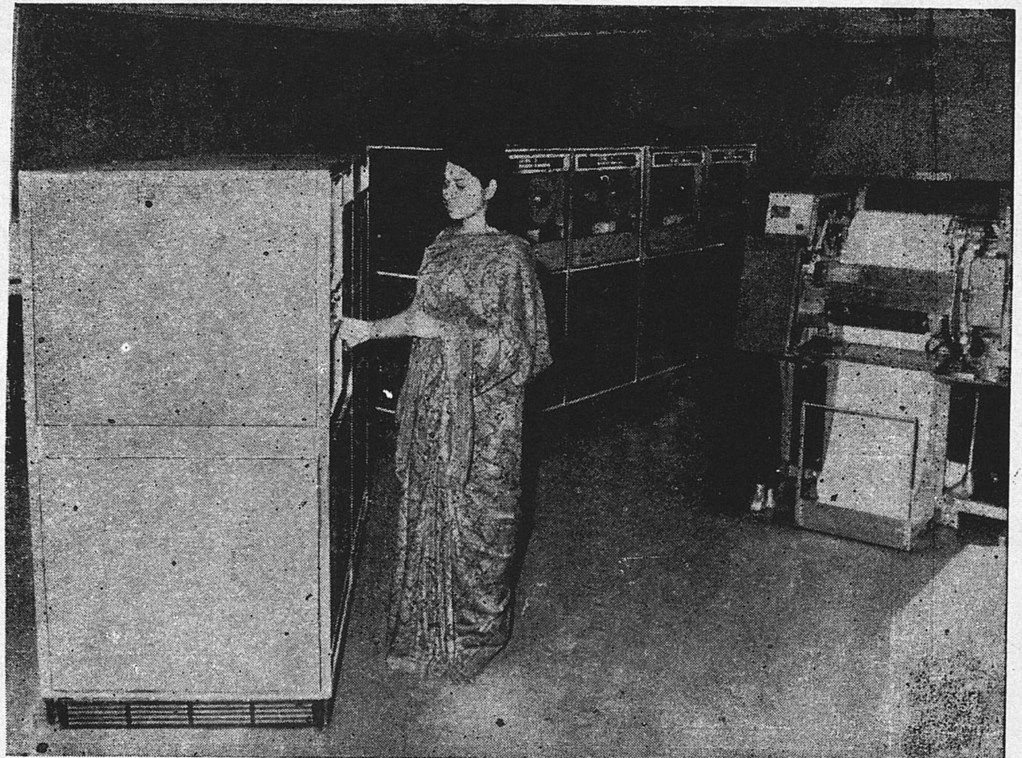
தமிழில் பயிலும் மாணவர்கள் உலக மாணவர்களிடையே சிறந்த இடம் பெறவேண்டுமென்பதே நம் குறிக்கோளாகும். கல்லூரிகளிலும் பல்கலைக் கழகங்களிலும், கலையியற் பாடங்களை யும், அறிவியற் பாடங்களையும், தொழில்நுட்ப அறிவுப் பாடங் களையும் பயிலுகின்ற மாணவர்கள், அவற்றைத் தமிழில் பயில வேண்டும் என்பதை வலியுறுத்தி வருவதற்குக் காரணம், தமிழறிவு வளரவேண்டும் என்பதைவிட, தமிழ் மக்களின் அறிவு ஆற்றல் எளிதாக விரைவாக வளரவேண்டும் என்பதுதான். 'எதிலும் தமிழ், எங்கும் தமிழ்' என்ற குறிக்கோளை நிறைவேற்ற வேண்டிய கடப்பாடு, தமிழகத்து ஆசிரியப் பெருமக்களையும் மாணவர்களையும் சார்ந்ததாகும். தமிழ்நாட்டுப் பல்கலைக் கழகங் களின் பல்வகை உதவிகளுக்கும் ஒத்துழைப்புக்கும் நம் மனம் கலந்த நன்றி உரியதாகுக.

இரா. நெடுஞ்செழியன்

பொருளடக்கம்

			பக்கம்
	கம்ப்யூட்டர் - அறிமுகமும் வரலாறும்	...	1
1.	கம்ப்யூட்டர் என்பது என்ன?	...	14
2.	கொள் - கொடு துறைகள்	...	50
3.	கம்ப்யூட்டர் ஒருங்கு	...	61
4.	ஒற்றைப் பதிவுப் பொறிகள்	...	87
5.	புரோகிராம் மொழிகள்	...	120
6.	கம்ப்யூட்டரின் பயன்கள்	...	153
7.	பிற	...	186
	Bibliography	...	207
	கலைச்சொற்கள்	...	208

குறிப்பு :— இந்நூலில் காணும் புகைப்படங்களுள் சில IBM, ICL ஆகிய நிறுவனங்களால் அளிக்கப்பட்டவை. மற்றவை கிண்டி பொறியியற் கல்லூரி கம்ப்யூட்டர் சென்டரில் எடுக்கப்பட்டவை. உதவிய இந் நிறுவனங்களுக்கு நன்றி உரித்தாகும்.



படம் 1
IBM 1401 கம்ப்யூட்டர்

கம்ப்யூட்டர் - அறிமுகமும் வரலாறும்

அறிமுகம்

மனிதன் ஆயிரக்கணக்கான பொருள்களைக் கண்டுபிடித்தான்; எத்தனையோ வகையான புதிய கருத்துகளையும் (Concepts) உருவாக்கினன். இவற்றுள் சில கண்டுபிடிப்புகளும் கருத்துகளும் மனித இனத்தின் நாகரிக வளர்ச்சிக்குப் பெரிதும் ஊக்கம் கொடுத்தன; சில, மனிதனின் வாழ்க்கை முறைகளையும், எண்ணப் போக்கையும் அடியோடு மாற்றியமைத்தன. வேறு சில கண்டுபிடிப்புகளோ, தகப்பன் சாமியாகத் தாம் எந்த அறிவின் பயனுகப் பிறந்தனவோ, அந்த அறிவின் பெரு வளர்ச்சிக்குக் காரணமாக ஆயின.

இன்றைய அறிவு வளர்ச்சியின் வேகமும் மனிதன் இதுவரை கண்டறியாததாகும். கடந்த இருபது ஆண்டுகளில் மனிதன் பெற்ற அறிவு வளர்ச்சி, அதற்கு முன்வரை அவன் தனது முழு வரலாற்றிலும் பெற்ற மொத்த அறிவு வளர்ச்சியைக் காட்டிலும் அதிகம். இந்த 'அறிவு' ஒவ்வோர் ஆண்டும் 10% அதிகரித்துக் கொண்டே போகிறதென்று கணக்கிட்டிருக்கிருர்கள்; ஒவ்வொரு பத்தாண்டுக் காலத்திலும் மனிதனின் அறிவு இரட்டிக்கிறது. இன்று ஆரும் படிவத்தில் படிக்கும் ஒரு மாணவனுக்கு உள்ள விஞ்ஞான அறிவு, பழங்கால கிரேக்கப் பேரறிஞர்களுக்கு இருந்ததைவிட அதிகம்; கல்லூரியில் முதலாண்டில் படிக்கும் விஞ்ஞான மாணவன், 'கலிலியோ'வுக்குத் தெரிந்ததைவிட அதிகப்படியான பௌதிக அறிவைப் பெற்றுள்ளான்.

இத்தகைய மாபெரும் அறிவு வளர்ச்சியால் பெருகிவரும் 'செய்தி வெள்ளத்தில்' மனிதன் தன்னைத்தானே மூழ்கடித்துக்

கொள்ளுவானே என்ற அச்சம் ஏற்பட்ட நிலையில், இன்றைய கம்ப்யூட்டர் அவனது உதவிக்கு வந்தது. இச் 'செய்தி வெள்ளத்தை' ஒழுங்குபடுத்தி, சேமித்து, உரிய துறையில், உரிய முறையில் பகிர்ந்தளிக்க மட்டும் கம்ப்யூட்டர் உதவவில்லை. அதுவே மனிதனின் அறிவு வளர்ச்சியின் வேகத்தை மேலும் அதிகரிக்கவும் துணை புரிந்தது ; துணை புரிகிறது.

அணுசக்தியும், விண்வெளிப் பயணமும் மனித நாகரிகத்தின் போக்கைப் பெரிதும் மாற்றிவிட்டன, மாற்றப் போகின்றன என்று கூறலாம். இத் துறைகளின் வளர்ச்சிக்குப் பெரிதும் உதவியா யிருந்தது, இருப்பது, கம்ப்யூட்டர் ஆகும். அணுசக்தியோ, ராக்கெட்டோ மனித நாகரிகத்தையும், அதன் எண்ணப்போக்கை யும் பாதித்ததை விட கம்ப்யூட்டர் பாதிக்கப்போகிறது என்று கூறினல் அது மிகையாகாது. மனிதனின் அன்றாட வாழ்வில் அணுசக்தியைப் பற்றியும், ராக்கெட்டைப் பற்றியும் அறிந்து கொள்ள வேண்டிய அவசியம் ஏதுமில்லை. ஆனல் இன்னும் சிறிது காலத்தில், தொழில், வணிகத் துறைகள் யாவும் கம்ப்யூட்டரைப் பயன்படுத்த ஆரம்பித்தவுடன் ஒவ்வொரு மனிதனும் கம்ப்யூட்ட ருடன் நேரிடையாகவோ, மறைமுகமாகவோ தொடர்பு கொண்டு தான் ஆகவேண்டியிருக்கும் ; இன்று மனிதன் எந்த அளவுக்கு மின்சாரத்தை அன்றாட வாழ்விற்கு நம்பியிருக்கிறனே அந்த அளவுக்கு அவன் வாழ்க்கை எதிர்காலத்தில் கம்ப்யூட்டருடன் பின்னிப் பிணைந்திருக்கும். இன்று ஒரு பெரும் நகரத்தில் மின்சாரம் தடைப்பட்டுப் போனல் எவ்வளவு குழப்பம் ஏற்படு கிறதோ, குறைந்தது அவ்வளவு குழப்பம் எதிர்காலத்தில் ஒரு கம்ப்யூட்டரின் இயக்கம் தடைப்படுவதால் ஏற்படக்கூடும்.

ஒவ்வோர் அடிப்படையான கண்டுபிடிப்பைக் கண்டும், மனித இனம் சற்றுப் பயந்து உள்ளது. முதன் முதலாக சக்கரத்தை (Wheel) ஆதிகால மனிதன் கண்டுபிடித்தபொழுது, கட்டாயம் அவனுக்கு எதிர்ப்பு ஏற்பட்டிருக்கும். ஒரு வகையில் இத்தகைய அச்சமும் எதிர்ப்பும் மனித இனத்தின் தற்காப்புணர் வின் எதிரொலி எனக் கொள்ளலாம். ஆனல் கம்ப்யூட்டர்கள் வந்தபொழுது ஏற்பட்ட அச்சம் ஓரளவு விசித்திரமானது. காரும், புகைவண்டியும் வந்தபோது கோச் வண்டிக்காரர்கள் தமது பிழைப்புப் போய்விடும் என்று எதிர்பார்த்திருப்பார்கள்; ஆனல் யாரும் மனிதனுக்குக் கால்கள் போய்விடும் என்று எண்ணவில்லை. கம்ப்யூட்டர் வந்தபோதோ காலப்போக்கில் மனிதன் தன் சிந்திக்கும் பழக்கத்தை விட்டு எண்ணும் திறனையே இழந்து விடுவான் என்றும், தொழிலாளிகள்

வேலைகளை இழந்துவிடுவர் என்றும் பயப்பட்ட மனிதர்கள் இருந்தார்கள். இத்தகைய அச்சத்திற்குக் காரணங்கள் இரண்டு; முதலாவது கம்ப்யூட்டர் என்ருல் என்ன? அது எவ்வாறு வேலை செய்கிறது? அதன் நிறை குறைகள் யாவை என்று பொதுமக்களுக்கு எடுத்துச் சொல்லுவார் யாருமில்லை. இந்த அறியாமைச் சூழ்நிலையில் மனம் போன போக்கில் உருவாக்கப் பட்ட விஞ்ஞானக் கற்பனைக் கதைகளும் (Science Fiction), திரைப் படங்களும், விவரம் அறியாத மக்கள் மனத்தில் அச்சத்தையும் 'கம்ப்யூட்டர்' என்ருல் அது ஏதோ ஒரு பயங்கர அரக்கன் என்ற எண்ணத்தையும் தோற்றுவித்ததில் வியப் பில்லை. நல்லவேளையாக இத்தகைய தவருன கருத்து இப்போது சிறிது சிறிதாக மறைந்து வருகிறது.

பார்க்கப்போனுல், மனிதன் இதுவரை கண்டுபிடித்த கருவி களிலேயே மிகவும் ஆற்றல் மிக்க கருவி கம்ப்யூட்டர் ஆகும். இன்று ஒரு மனிதன் பித்தான் ஒன்றை அழுத்தி தனது உடல் ஆற்றலைப் போலப் பல்லாயிரம் மடங்கு ஆற்றல் படைத்த இயந்திரங்களை இயக்குகிருன். அதேபோல் கம்ப்யூட்டரில் ஒரு பித்தானை அழுத்தி ஒரு மனிதனின் மூளையாற்றலைப்போல் பன் மடங்கு ஆற்றல் வாய்ந்த கம்ப்யூட்டரின் உழைப்பைப் பயன் படுத்திக் கொள்கிருன்; அவ்வளவே.

கம்ப்யூட்டர்கள் ஏற்கெனவே மனிதனின் மனப்போக்கில் மாறுதல்களை ஏற்படுத்த ஆரம்பித்துவிட்டன. விஞ்ஞானம் பொறியியல் துறைகளில் 'பரிசோதனைகள்' (Experiments) மிகவும் முக்கியம் வாய்ந்தவை. சில துறைகளில் பரிசோத னைக்குத் தேவையான பொருள்கள் விலைமதிப்பு வாய்ந்தவைகளா யிருக்கலாம். அல்லது பரிசோதனைக்கான ஆயத்தங்கள் செய் வதற்கே மிகவும் பணச் செலவும், காலச் செலவும் ஆகலாம். கம்ப்யூட்டர்கள் வந்த பிறகு சோதனைச்சாலைகளில் (Laboratories) பரிசோதனைகள் குறைய ஆரம்பித்துள்ளன. இதற்குக் காரணம் புதிதாகத் தோன்றிய போலி நிகழ்வு (Simulation) ஆய்வு முறை யாகும். இம் முறை கம்ப்யூட்டர் தந்த முறையாகும். கருத்தள வில் இம் முறை கம்ப்யூட்டர் வருவதற்கு முன்பே யிருந்திருப் பினும், நடைமுறையில் அதைப் பயனுள்ளதாக ஆக்கியது கம்ப்யூட்டரே. இம் முறையால் நாம் செய்ய விரும்பும் விஞ் ஞானப் பரிசோதனைகளைக் கம்ப்யூட்டருக்குள் வெறும் எண்களைக் கொண்டு நிகழ்த்திக் காட்டலாம். இன்றைய விண்வெளிப் பயணம் (Manned space flight) ஒவ்வொன்றும் தொடங்குவதற்கு முன்னர் நூற்றுக்கணக்கான போலிப் பயணங்கள் கம்ப்யூட்டரின்

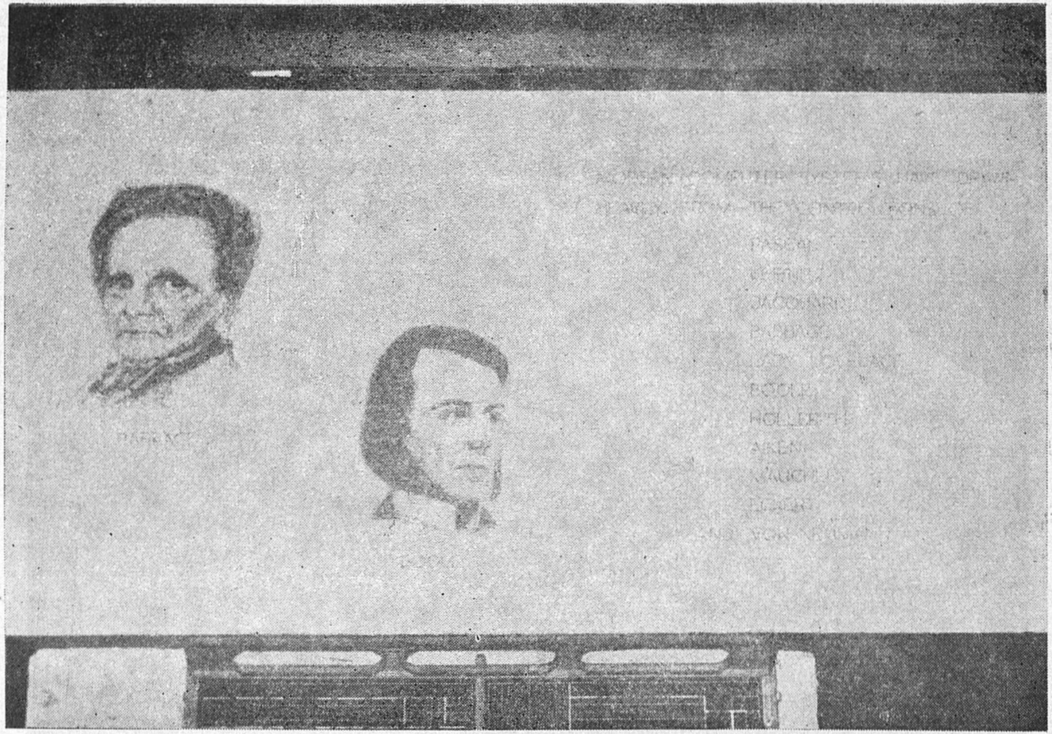
துணையுடன் மேற்கொள்ளப்படுகின்றன. இதனுல் உண்மையான பயணம் தொடங்கும்பொழுது பயணிகள் யாவரும், பெருமளவில், முன் அனுபவம் பெற்றவர்கள்போல் உணருகிருர்கள். ஒருவரை ஆயிரம் தடவை புகைப் படத்தில் பார்த்த பிறகு நேரில் பார்க்கும் போது ஏற்படும் அனுபவமே இப் பயணிகளுக்கு, பல போலிப் பயணங்களுக்குப் பின்னர், உண்மையான பயணம் செய்யும் போது ஏற்படுகிறது. இன்றுள்ள போலி நிகழ்வு ஆய்வு முறை **(Simulation technique)** இன்னும் முழுமை யடைந்து பெருமள வுக்குப் பயன்படுத்தப்படும் நிலை ஏற்பட்ட பிறகு பௌதிகம், பொறியியல், இரசாயனம், மருத்துவம் போன்ற துறைகளில் பரிசோதனை செய்ய விழைவோர், பரிசோதனைச்சாலைக்குச் செல்லமாட்டார்கள். அதற்கு மாருகக் கம்ப்யூட்டர் இருக்கு மிடத்தைத் தேடிப் போவார்கள்.

கம்ப்யூட்டரோடு தொடர்பு ஏற்பட, ஏற்பட மனிதனின் எண்ணத்தில் கச்சிதமும் கட்டுப்பாடும் ஏற்படுகின்றன. இதைக் கம்ப்யூட்டருடன் நேரிடைத் தொடர்பு கொண்ட ஒவ்வொருவரும் அறிவர். எண்ணத்தெளிவும், செய்முறை பற்றிய முழு அறிவும் இல்லாமல் எந்த புரோகிராமாவது **(programme)** எழுதப்படுமாளுல் அதற்கான தண்டனை—கம்ப்யூட்டரால் கொடுக்கப்படும் தண் டனை—கடுமையானதாயிருக்கும்; கம்ப்யூட்டரிலிருந்து கிடைக்கும் விடை பைசாவிற்குப் பயன்படாது. இந்த எண்ணக் கட்டுப்பாடு காரணமாகக் கம்ப்யூட்டரைப் பயன்படுத்துவோரின் வெளியுலகக் கண்ணேட்டத்திலும்கூட மாறுதல் ஏற்படுகிறது. சுற்றி உள்ள உலகில் **என்னென்ன** நிகழ்ச்சிகள் நடக்கின்றனவென்று அவர்கள் காண்பதில்லை; நடக்கும் ஒவ்வொரு நிகழ்ச்சியும் எவ்வாறு நடக்கிறதென்று அவர்களது மனம் தேட ஆரம்பிக்கிறது.

ஒரு காலத்தில் பள்ளிக்கூடங்களில் கம்ப்யூட்டர்கள் பெரும் அளவில் பயன்படுத்தப்படத்தான் போகின்றன. இன்றுள்ள பள்ளி மாணவனுக்கு ஓர் அடிக்கோல், செட்ஸ்கொயர் ஆகிய வற்றில் எவ்வளவு பழக்கம் உண்டோ, அவ்வளவு பழக்கம் வருங் காலத்தில் கம்ப்யூட்டர்களுடன் ஏற்படத்தான் போகிறது. அப்போது எந்த நன்மை ஏற்பட்டாலும், ஏற்படாவிட்டாலும், மாணவர்களது எண்ணத்தில் ஒழுங்கு, கட்டுப்பாடு, கச்சிதம் ஆகியவை தாமாக ஏற்படும். தெளிந்த சிந்தனை பெற்ற மனிதர் களைவிட ஒரு சமூகத்திற்கு வேறென்ன வேண்டும் ?

வரலாறு

இன்று நாம் கம்ப்யூட்டர் என்று குறிப்பிடும் பொறியின் வயது 25 ஆண்டுகளுக்குள்தான். ஆனுல் இதன் முன்னேடி



படம் 0.1 கம்ப்யூட்டர் உலகின் காரணர்கள்: சார்லஸ் பாபேஜ் — ஜார்ஜ் பூல்

சார்லஸ் பாபேஜ்: ஆங்கிலக் கணித மேதையான இவர் பலவேறு துறைகளிலும் சிறந்து விளங்கினர். அவரது கண்டுபிடிப்புகளுக்குள் தலைசிறந்தனவாகக் கருதப்படுவன, அவர் வடிவமைந்த அனலடிக்கல் எஞ்சின், டிபரன்ஷியல் எஞ்சின் என்ற இரண்டு பொறிகளுமே. அவற்றின் வழித் தோன்றல்களே இன்றைய கம்ப்யூட்டர்கள்.

ஜார்ஜ் பூல்: பெரிதும் வறுமையில் வாடிய இக் கணித மேதை ஒரு தனிப்பட்ட 'அல்ஜிப்ரா'வையே உருவாக்கினர். பைனரி எண்களைக் கொண்டே உருவாக்கப்பட்ட இவரது 'பூலியன் அல்ஜிப்ரா' இன்று எலக்ட்ரானிக்ஸ் (குறிப்பாக கம்ப்யூட்டர்) துறையின் உயிர்நாடியாகும்.

கி.மு. 2000 ஆண்டுகளுக்கும் முன்னர் பயன்படுத்தப்பட்ட மணிச்சட்டம் (Abacus) ஆகும். இன்று நம் ஆரம்பப் பள்ளிச் சிறுவர்கள் பயன்படுத்தும் இச்சட்டத்தைப் பயன்படுத்தி கிட்டத் தட்ட 5000 ஆண்டுகளுக்கு முன்னரே சீனர்கள் கணக்குகளைச் செய்தனர். இந்த மணிச் சட்டத்தில் ஆரம்பித்து, அப்போதைக் கப்போது புதிது புதிதாகக் கணிப் பொறிகள் கண்டுபிடிக்கப் பட்டுப் பயன்படுத்தப்பட்டன.

இன்றைய கம்ப்யூட்டரின் அமைப்பு, பாஸ்கல், லீப்னிட்ஸ், ஜெக்கார்ட், பாபேஜ், லவ்லேஸ், ஹாலரித், ஐக்கன், மாச்லே, எக்கர்ட், நாய்மன் என்பவர்களுக்குப் பெருமளவுக்குக் கடன் பட்டுள்ளது. இந்தப் பத்து மனிதர்களின் கண்டுபிடிப்புகளும், கருத்துகளுமே இன்றைய கம்ப்யூட்டரின் உயிர் நாடியாக உள்ளன.

பிலெய்ஸ் பாஸ்கல் ஒரு பெரிய கணித நிபுணர். தமது 19ஆம் ஆண்டிலேயே ஒரு கணிப்பொறியைச் செய்த மேதை. இவருக்கு, அடுக்கடுக்காக எண்களைக் கூட்டிக் கழித்துச் செய்வதற்குப் பொறுமையில்லை; அது வீண் காலச்செலவு என்பதும், அந்த நேரத்தில் வேறு பயனுள்ள சிந்தனையில் ஈடுபடலாம் என்பதும் அவர் எண்ணம். குறிப்பாக பாஸ்கலின் தந்தை ரூன் நகரின் நிர்வாகியானபின் நகர வரவு செலவு சம்பந்தப்பட்ட பெரும் கூட்டல் கணக்குகளைச் செய்யவேண்டிய பொறுப்பை பாஸ்கலிடம் ஒப்படைத்தார். இந்த வேலையின் அலுப்பைத் தாளாமல் பாஸ்கல் கணிப்புப் பொறி ஒன்றைக் கண்டுபிடிப்பதில் முனைந்து 1642ஆம் ஆண்டில் வெற்றியும் கண்டார். இவரது பொறியில் பத்துச் சக்கரங்கள் இருந்தன. ஒவ்வொரு சக்கரமும் 0 முதல் 9 வரையிலான இலக்கத்திற்கானது. இந்தச் சக்கரங்களும், இவற்றை இயக்குவதற்கான பற்சக்கரங்களும் (Gears) சேர்ந் ததே பாஸ்கலின் பொறி ; இப் பொறி இன்றைய கூட்டல் பொறியை (Adding machine) பெரிதும் ஒத்திருந்தது.

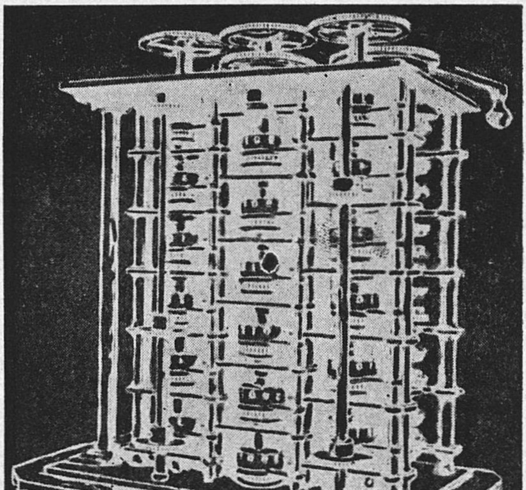
பாஸ்கலுக்குப் பின்னர் வந்த காட்பிரைட் வில் ஹெல்ம் லீப்னிட்ஸ் பலவேறு துறைகளிலும் ஈடுபாடு கொண்டிருந்தவர். கணிதம், பௌதிகம், கடல் பயணத்துறை (Nautical Sciences) ஆகிய பலவற்றிலும் வியத்தகு கண்டுபிடிப்புகளைச் செய்த லீப்னிட்ஸ், பாஸ்கலின் கணிப்புப் பொறியை மேலும் விஸ்தரித்து ஒரு புதிய பொறியைச் செய்தார். பாரிஸ் அகடெமியிலும், லண்டன் ராயல் சொசைடியிலும் இயக்கிக் காட்டப்பட்ட இப் பொறி, கூட்டல் மட்டுமல்லாமல் கழித்தல், பெருக்கல்,

வகுத்தல், வர்க்கமூலம் ஆகியவற்றையும் செய்தது. இதை இன்றைய 'டெஸ்க் கால்குலேட'ருடன் முற்றும் ஒப்பிடலாம்.

லீப்னிட்ஸுக்கு அடுத்து இவ்வரிசையில் வருபவர் இன்றைய நவீன கம்ப்யூட்டர்களின் தந்தையான சார்லஸ் பாபேஜ் ஆகும். லீப்னிட்ஸுக்கு ஒன்றரை நூற்றுண்டுகளுக்குப் பின்னர் தோன்றிய இந்த ஆங்கிலக் கணித வல்லுநரின் ஆர்வமும் பல துறைகளிலும் இருந்தது. தமது காலத்தில் இங்கிலாந்திலிருந்த கணிதக் கல்வியில் பெரிதும் அதிருப்தியுற்ற பாபேஜ், கணிதக் கல்வியில் குறிப்பாக நுண்ணளவு கணிப்புத் (Infinitesimal Calculus) துறையில் லீப்னிட்ஸின் குறியீடுகளைப் (Symbols) பின்பற்றுமாறு சிபாரிசு செய்தார்.

பாபேஜின் கண்டுபிடிப்புகளும் பலதிறப்பட்டவை. பனிப் பாறை ஆராய்ச்சி, சீரான தபால் கட்டணம் (uniform postage rate) பார்சல் தபால், கடலடிப்பயணம், கிரீன்விச் நேர அறிவிப்பு முறை, மின்காந்த சுழற்சிகளின் காரணம் ஆகிய முற்றிலும் மாறுபட்ட துறைகளில் அவரது ஆய்வு பயனுள்ள பல முடிவுகளை ஈன்றது. இவை மட்டுமில்லாமல், ஸ்பீடாமீட்டர், புகைவண்டியில் எஞ்சின் முன்னால் இருக்கும் கௌ காட்சர் (Cow Catcher) முதலியனவும் அவரது கண்டுபிடிப்புகளே. 'மெஷின் டூல்' உற்பத்தி, பேருற்பத்தி முறை (Mass Production) இவற்றையும் பெரிதும் ஊக்குவிக்க வேண்டும் என்று வாதாடிய பாபேஜ் இன்றைய 'ஆபரேஷன் ரிசர்ச்' (Operation Research) துறையின் முன்னோடியும் ஆவார். இவ்வாராய்ச்சி முறையைக் குண்டூசித் தொழிற்சாலைகளிலும், புத்தகம் பதிப்பிக்கும் தொழிலிலும் முதன் முதலாகக் கையாண்டு பாபேஜ் வெற்றி கண்டார்.

இத்தனைக் கண்டுபிடிப்புகளைக் காட்டிலும் பாபேஜுக்குப் பெயர் வாங்கித் தந்தவை வேறு இரண்டு பொறிகளே அவைகளே இன்றைய கம்ப்யூட்டர்களின் முன்னோடிகளாம். அவற்றில் ஒன்றிற்கு 'டிஃபரன்ஸ் எஞ்சின்' (Difference Engine) என்று பெயர். ஓரளவுக்கு இந்தப் பொறியின் வடிவமைப்பிற்கு கி.பி. 1786ஆம் ஆண்டு 'முல்லர்' (ஜெர்மானியர்) என்பார் வடிவமைத்த ஒரு பொறி உதவியாயிருந்திருக்கலாம். ஆனால் முல்லர் தமது பொறியைச் செயல்படும் வகையில் செய்யவே யில்லை; அதுபோலவே பாபேஜும் தமது பொறியைச் செய்ய முற்பட்டாலும் அதை வெற்றிகரமாகச் செய்து முடிக்கவில்லை. அதற்குக் காரணம் அன்றைய விஞ்ஞானம், பொறியியல் துறைகள் அவருக்குத் தேவையான அளவு முன்னேறியிருக்க



படம் 0.3

பாபேஜின் டிபரன்ஸ் எஞ்சின்

வில்லை என்பதேயாகும். பாபேஜ் பாவம், நூறு ஆண்டுகள் முன் கூட்டிப் பிறந்துவிட்டார்.

பாபேஜின் இரண்டாவது பொறி 'அனலெடிக்கல் எஞ்சின்' ஆகும். மற்ற எல்லாவற்றையும் விட பாபேஜின் பெருமையை நினைவுகூர்வது இப் பொறியே. இன்றைய விஞ்ஞான, தொழில் நுட்ப வசதிகள் மட்டும் அவர் காலத்திலிருந்து, அவர் கனவு கண்ட பொறியை உருவாக்க முடிந்திருந்தால் 1822ஆம் ஆண்டே இன்றைய கம்ப்யூட்டர் பிறந்திருக்கும். தமது கனவுப் பொறியைப் பற்றிய ஆயிரக்கணக்கான வரைபடத்தை (Drawings) பாபேஜ் வரைந்திருக்கிருர். அன்று அவற்றைப் புரிந்துகொண்டோர் ஓரிருவரே. பாபேஜின் 'அனலடிக்கல் எஞ்சினின்' அடிப்படையும் இன்றைய கம்ப்யூட்டரின் அடிப்படையும் ஒன்றே.

பாபேஜின் காலத்தில் அவரைக் கேலி செய்தவர்கள் பலர்; அவரது பொறியால் யாருக்கும் எந்தப் பயனும் இல்லை என்று பழித்தோரும் இருந்தனர். 'பாபேஜின்' மடமைச் சின்னம் என்று அவரது 'அனலடிக்கல் எஞ்சினுக்குப்' பெயரும் சூட்டினர். எந்தத் துறையிலும் புதுமை விளைவிக்க விரும்பு வோர்க்கு, மனித இனத்திடம் அளவற்ற பொறுமையும், சகிப்புத் தன்மையும் வேண்டும்; துரதிர்ஷ்டவசமாக பாபேஜிடம் இந்த இரண்டுமில்லை.

பாபேஜிற்கு ஏற்பட்ட சங்கடங்கள் பல. தெருப் பாடகர் களால் அவர் சிரமப்பட்டார். அவரது அறைக்கு நேரே கீழே நின்றுகொண்டு அவர்கள் பாடிக்கொண்டிருப்பார்கள். பாபேஜின் சிந்தனைத் தொடர் சிதைக்கப்படும்; ஆத்திரத்துடன் பாபேஜ் பாடகர்களைக் கோபித்துக் கொள்ளுவார். அதற்கு அவர்களது பதில் இன்னும் பலரைச் சேர்த்துக்கொண்டு முன்னைக் காட்டிலும் சத்தமாகப் பாடுவதுதான். இத்தகைய தொல்லைகள் மட்டுமல்லாது தனது சகாக்களுடனும் அவருக்கு எப்போதும் மோதல் ஏற்பட்டுக்கொண்டேயிருந்தது; குறிப்பாக அவரது காலத்து அஸ்ட்ரானமர் ராயலுடன் அவருக்குக் கருத்து வேறுபாடு இருந்தது. அப்போது பயன்படுத்தப்பட்டுவந்த விண்கோள் பட்டியலும் (Astronomical Table) கணிப்புப் பட்டியலும் (Mathematical Table) நிறைய தவறுகளைக் கொண்டிருந்தன. இத் தவறுகள் பாபேஜிற்குப் பெரிதும் எரிச்சலூட்டின. இத் தவறுகள் ஏதும் இல்லாமல் சரியான பட்டியலைத் தயாரிக்க வென்றே பாபேஜ் தமது அனலடிக்கல் எஞ்சினைத் தயாரிக்க முற்பட்டார்.

இருப்பினும் எதிர்காலத்தில் கம்ப்யூட்டரின் பயன்களைப் பற்றி அவருக்குத் தெளிந்த அறிவு இருந்திருக்கிறது. வான நூலைப் போலவே, இரசாயனத் துறையிலும், பரிசோதனைகளிலிருந்து தேவையான அடிப்படை விவரங்களை எடுத்துக்கொண்டு, அவற்றைக் கணிதச் சமன்பாடுகளில் பயன்படுத்திப் புதிய புதிய கூட்டுப் பொருள்களின் (Compounds) குணங்களை ஆராய்வதுடன், அவை எந்தெந்த மூலப் பொருள்களால் ஆக்கியிருக்கக் கூடும் என்ற விவரத்தையும் பரிசோதனை ஏதுமின்றி, வெறும் கணிப்பு மூலமே அறியக்கூடும் என்று ஒரு சமயம் தீர்க்கதரிசி போல், பாபேஜ் கூறினுர். அதற்கு 115 ஆண்டுகளுக்குப் பின்னர் ஆய்வு நிலையம் ஒன்றில் விட்டமின் B_{12}-ன் படிக அமைப்பைக் கம்ப்யூட்டர் ஒன்று கணித்துக் கொடுத்தது.

பாபேஜின் அனலடிக் எஞ்சினை முற்றிலும் புரிந்துகொண்ட ஒரே நபர் லேடி லவ்லேஸ் (Lady Lovelace) ஆகும். ஆங்கிலக் கவி லார்ட் பைரனின் மகளான லவ்லேஸ், நல்ல கணிதஞானம் படைத்தவர். பாபேஜின் பொறியில் செய்வதற்கென்றே பல கணக்குகளைச் சேர்த்து வைத்திருந்தார்; உலகின் முதல் கம்ப்யூட்டர் புரோகிராமர் என்று கூறப்படத்தக்க இவர் பாபேஜின் கணிப்பில் இருந்த ஒரு தவறை எடுத்துக் காட்டி அவரது பாராட்டைப் பெற்ருர்.

பாபேஜின் எஞ்சின் இன்றைய கம்ப்யூட்டர்களைப் போலவே புரோகிராம் ஏற்கவல்லதாகவும், புரோகிராமின் ஒரு பகுதியிலிருந்து வேறு ஒரு பகுதிக்குப் பிரிந்து செல்லவல்லதாயு மிருந்தது; இவற்றிற்கு மேலாக, பாபேஜின் எஞ்சினும் இன்றைய கம்ப்யூட்டர்களைப் போலவே துளைக்கார்டுகளைப் பயன்படுத்த வல்லதாயும் அமைந்திருந்தது.

கம்ப்யூட்டர் உலகில் கார்டுகளை முதலில் புகுத்தியவர் ஹெர்மன் ஹாலரித் ஆகும். இவர் கம்ப்யூட்டருடன் நேரிடைத் தொடர்பு கொண்டில்லாதவராயிருப்பினும், இவர் தயாரித்து விற்பனை செய்த பொறிகளுக்குக் கார்டுகள் தேவைப்பட்டன. இக் கார்டுகளே பின்னர் கம்ப்யூட்டர்களிலும் பயன்படுத்தப் பட்டன.

1925ஆம் ஆண்டு, (MIT—Massachussettes Institute of Technology)-யில் டாக்டர் 'வன்னிவர் புஷ்' என்ற விஞ்ஞானி ஒரு பெரிய கம்ப்யூட்டரைச் செய்து முடித்தார். 'டிஃபரென்ஷியல் அனலைசர்' (Differential Analyser) என்று அழைக்கப்

பட்ட இப் பொறி ஏராளமான பற்சக்கரங்களைக் கொண்டு இயங்கியது. பாபேஜின் பொறியைப் போலவே ‘புஷ்’ஷின் பொறியும் முற்றும் இயந்திர சக்தியால் (Mechanical Power) இயங்கியது.

வணிகத்துறையில் பயன்படுத்தப்பட்டுவந்த கணிப்புப் பொறிகளை, விஞ்ஞானக் கணிப்புகளுக்கு முதன் முதலில் பயன் படுத்திய பெருமை ‘டாக்டர் எல். ஜே. கோம்ரீ’யைச் சாரும். 1926ஆம் ஆண்டு கிரீன்விச் விண்ணோக்கு நிலையத் தலைவரா யிருந்த கோம்ரீயும் பாபேஜைப் போலவே கைக்கணிப்பில் (Hand Computation) ஏற்பட்டுக் கொண்டிருந்த தவறுகளால் பெரிதும் அயர்ச்சியுற்றார். பரோ போஸ்ட்டிங் மெஷின் (Burrough Posting Machine) என்ற கணிப் பொறி ஒன்றைப் பெற்று கோம்ரீ தமது கணிப்புகளுக்கு அதைப் பயன்படுத்த ஆரம்பித்தார். இப் பொறியில் பயன்படுத்துவதற்காக 500,000 கார்டுகளை கோம்ரீ துளையிட வேண்டியிருந்தது. கோம்ரீயின் இக் கணிப்பின் பயனாக வெளிவந்தது ஓர் அரிய பெரிய அட்டவணை. இவ்வட்டவணையில் கோம்ரீயின் காலத்திலிருந்து கி.பி. 2000 ஆண்டுவரை ஒவ்வொரு நாளுக்கும் நண்பகல், நள்ளிரவு நேரங்களில் நிலவின் இருப்பிடம் கணித்துக் கொடுக்கப் பட்டிருந்தது. இந்த அட்டவணை கடல் பயணத்திற்கு மிக மிக இன்றியமையாததாய் அமைந்தது.

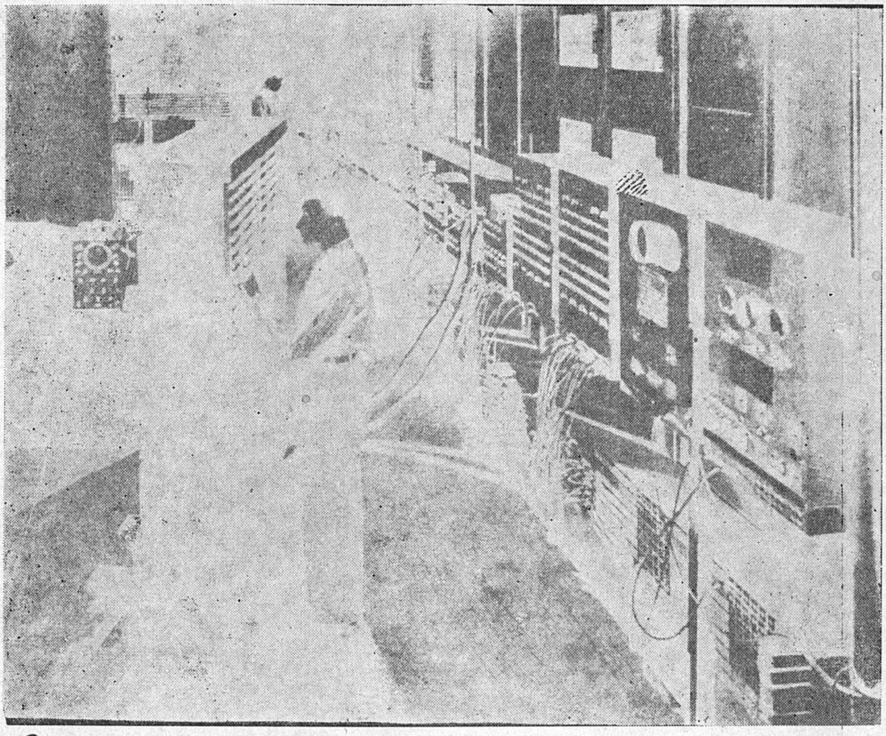
கோம்ரீயின் இந்த அரிய சாதனை, அதுவரை எளிய வணிகத் துறைக் கணிப்புகளுக்கு மட்டுமே பயன்படுத்தப்பட்டு வந்த கணிப் பொறிகளை, கம்ப்யூட்டர்களை, விஞ்ஞானக் கணிப்புகளுக் கும் பயன்படுத்த இயலும் என்பதை நிரூபித்தது.

இதேசமயம், அமெரிக்காவில் கொலம்பியா பல்கலைக் கழகத்தின் விண்ணியல் பேராசிரியராக ‘டாக்டர் டபிள்யூ. ஜே. எக்கர்ட்’ என்பவர் இருந்தார். கோம்ரீயின் சாதனையைக் கேள்வியுற்ற இப் பேராசிரியர், தாமும் அதுபோல் செய்ய எண்ணி ஐ.பி.எம். (International Business Machines) கம்பெனியைத் தமக்கு வேண்டிய கருவிகளைச் செய்து கொடுக்குமாறு கேட்டுக் கொண்டார். எக்கர்ட் பயன்படுத்திய இக் கம்ப்யூட்டர் ஒரு மின்-இயந்திரப் (Electromechanical) பொறியாகும்.

முதன் முதலில் தயாரிக்கப்பட்ட ஆடோமாடிக் கம்ப்யூட்டர் ‘மார்க்-I’ ஆகும். இதன் தந்தை ‘டாக்டர் ஹோவார்ட் ஐக்கன்’ என்ற ஹார்வார்டு பல்கலைக் கழகக் கணிதப் பேராசிரி யர்; இக் கம்ப்யூட்டரைச் செய்தது ஐ.பி.எம். இது செயல்பட

ஆரம்பித்தது 1944ஆம் ஆண்டு. கம்ப்யூட்டர் வரலாற்றில் ஒரு மைல் கல்லாய் அமைந்த இந்தப் பொறி பாபேஜின் கனவை நனவாக்கிய முதல் கருவியாகும். பாபேஜின் அனலடிக் எஞ்சினைப்பற்றி முதலில் ஐக்கன் ஏதும் அறிந்திருக்கவில்லை. தமது பொறியின் வடிவமைப்பில் கிட்டத்தட்ட 3 ஆண்டுகள் செல விட்ட பிறகே அவருக்கு பாபேஜின் எஞ்சினைப்பற்றித் தெரிய வந்தது. பாபேஜின் எஞ்சினைப் பெரிதும் ஒத்தே ஐக்கனின் பொறியும் அமைந்திருந்தது. என்றாலும் ஓரிரு வேறுபாடுகள் இருக்கத்தான் செய்தன. எடுத்துக்காட்டாக ஐக்கனின் பொறி யில் புரோகிராம் ஆணைகளில் பிரிந்துபோகும் (**Branching Facility**) வசதி இல்லை. (பாபேஜின் அனலடிக் எஞ்சினில் பிரிந்து செல்ல வசதியிருந்தது என்று நாம் ஏற்கெனவே பார்த்தோம்.) இருப் பினும் இவ் வசதி பிற்காலத்தில் ஐக்கனின் பொறியிலும் சேர்க் கப்பட்டது.

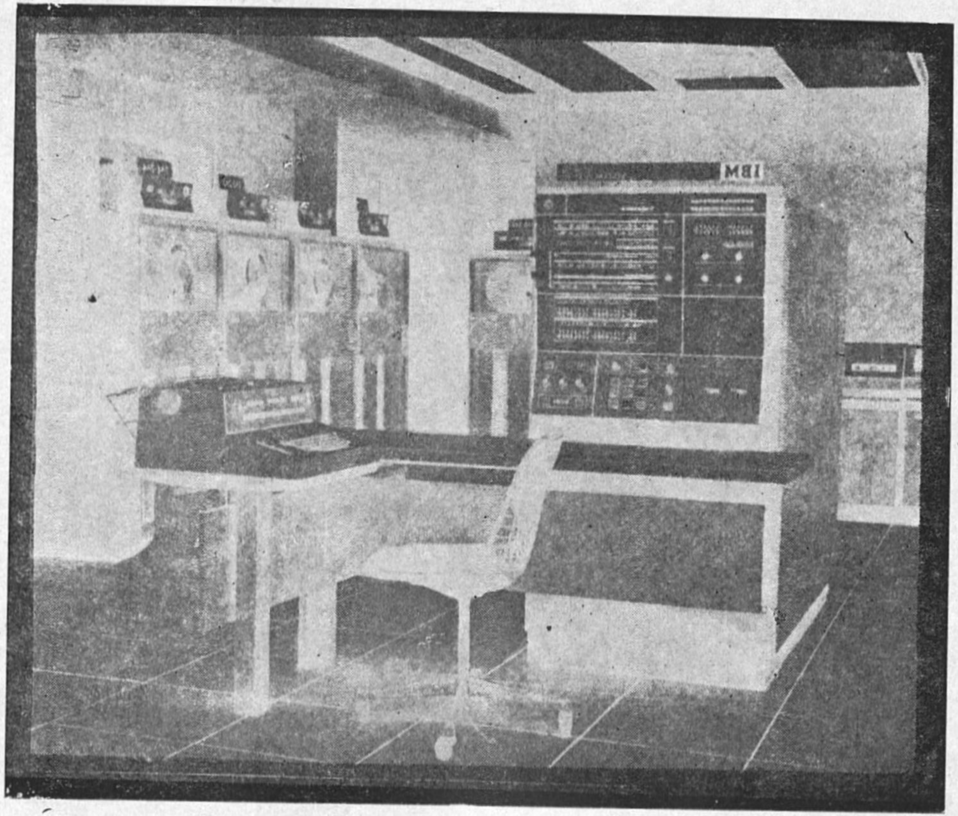
ஐக்கனின் இப் பெரிய கம்ப்யூட்டரின் பொறி முகம் (**Panel**) மட்டும் 51அடி நீளமும் 8அடி உயரமும் கொண்டிருந்தது. இன்றைய கம்ப்யூட்டரைப் போலவே அதில் ஒரு நினைவகமும் (**Storage**) இருந்தது. இந்த நினைவகத்தில் எண் வடிவ விவ ரங்கள் (**numeric data**) மட்டுமே இருத்திவைக்க வசதியிருந்தது. பெயர்களையோ அல்லது மற்றக் குறியீடுகளையோ இதன் நினைவகத் தில் இருத்திவைக்க வழியில்லை. புரோகிராம் சம்பந்தப்பட்ட கட்டளைகள் வெளியிலிருந்து துளைக்குறிகள் (**Punched Codes**) கொண்ட காகித நாடாவால் கொடுக்கப்பட்டன. இந்த நாடா ஒரு நிமிடத்திற்கு 200 தடவை என்ற வேகத்தில் தத்தித்தத்தி (**Moved in Jerks**) நகர்ந்துகொண்டிருந்தது. இதன் வேகம், டெஸ்க் கால்குலேட்டர் உதவியால் 20 மனிதர்கள் கணிக்கும் கணிப்பிற்கு ஈடாகும். மார்க் I-ன் பரம்பரையில் வந்த மார்க்-III வால்வுகளையும், மார்க்-IV ட்ரான்ஸிஸ்டர்களையும் கொண்டு செய்யப்பட்டன. எனினும் மார்க் பரம்பரை கம்ப்யூட்டர்கள் நிலைத்துநிற்க முடியவில்லை. இதன் முக்கிய காரணம், கம்ப்யூட் டரில் புரோகிராம் ஆணைகளை வெளியிலிருந்து காகித நாடாக்கள் மூலம் கொடுக்கவேண்டியிருந்ததே. இன்றைய கம்ப்யூட்டர் களில் உள்ளதைப் போல் ஆணைகளை ஏற்று, நினைவிலிருத்திக் கொண்டு பின்னர் செயல்படும் தன்மை அவற்றில் இல்லை. 'மார்க்' கம்ப்யூட்டர்களுக்குப் பின் வந்தவை புரோகிராம் ஆணை களை உள்ளடக்கி அதன் பின்னர் அவ்வாணைகளை நிறைவேற்றும் ஆற்றல் பெற்றிருந்தன. இக் கம்ப்யூட்டர்கள் 'மார்க்' கம்ப்யூட் டர்களைக் காட்டிலும் கையாள, பல விதங்களிலும் சௌகரிய மானவை; கணிப்பு வேகம் மிக்கவை.



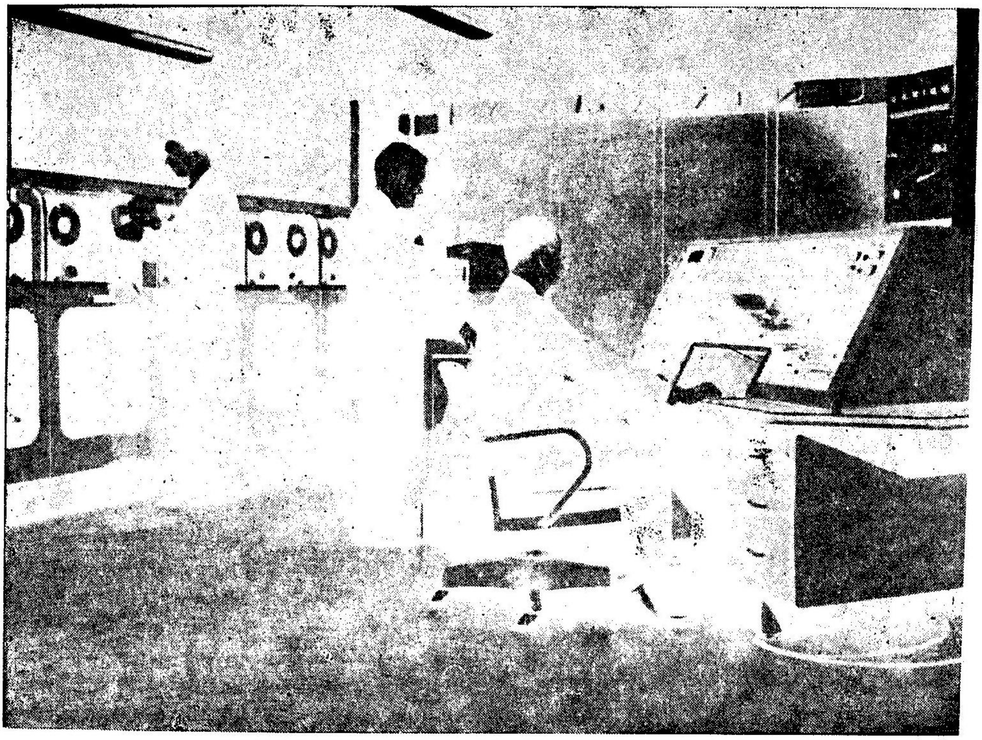
படம் 0.4

கம்ப்யூட்டர் எனியாக் (ENIAC)

ENIAC என்பது Electronic Numerical Integrator and Computer என்பதன் சுருக்கம். எலக்ட்ரானிக் பகுதிகளைக் கொண்டு செய்யப்பட்ட முதல் கம்ப்யூட்டர் ENIAC ஆகும். இதற்குமுன் வடிவமைத்துச் செய்யப்பட்டவை யாவும் பெருமளவுக்கு, பற்சக்கரங்கள், நெம்பிகள் போன்ற இயந்திர பாகங்களைக்கொண்டு செய்யப்பட்டவை.

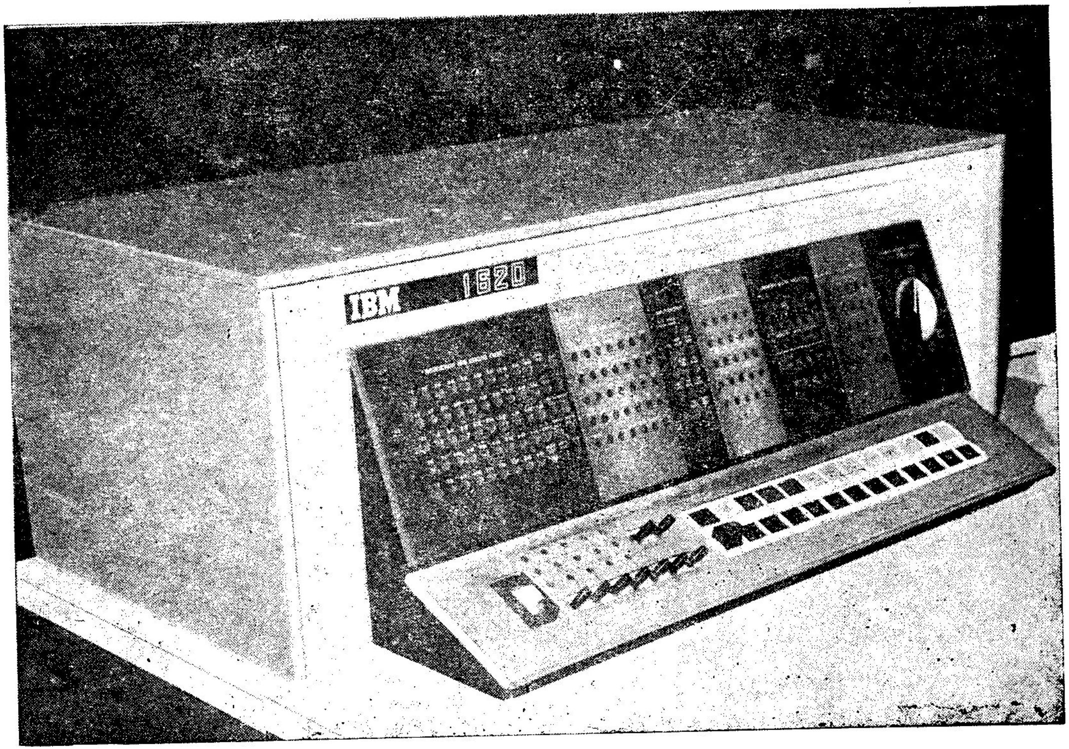


படம் 0.4a
யூனிவாக் (UNIVAC) கம்ப்யூட்டர்



படம் 0.4b

IBM 360 கம்ப்யூட்டர்



படம் 0.4c

IBM 1620 கம்ப்யூட்டரின் முகப்புத் தோற்றம்

கம்ப்யூட்டர்களின் இயக்கம் யாவும் ஆயிரக்கணக்கான ஸ்விட்ச்களால் நடைபெறுகிறது. மார்க்-I கம்ப்யூட்டரின் ஸ்விட்ச்கள் யாவும் மெகானிக்கல் ஸ்விட்ச்களே; மெகானிக்கல் ஸ்விட்ச்களின் இயக்க வேகம், மிகவும் குறைவு. அதனுல் மார்க்-I-ன் வேகமும் குறைவாகவே இருந்தது. மார்க்-Iக்குப் பின்னர் வந்த 'எனியாக்' (ENIAC) கம்ப்யூட்டர்தான் முதல் எலக்ட்ரானிக் கம்ப்யூட்டர் ஆகும். இதைச் செய்தவர்கள் 'டாக்டர் ஜான் மாச்லே' (Dr. John Mauchley) யும் அவரது ஆராய்ச்சி மாணவரான 'பிரஸ்பர் எக்கர்ட்டும்' ஆவர். 'எனியாக்' செயல்படும் நிலைக்கு வருவதற்கு முன் எத்தனையோ தடைகளும் துன்பங்களும் நேரிட்டன. மாச்லே எலக்ட்ரானிக் கம்ப்யூட்டர் ஒன்றை வடிவமைத்துப் பென்சில்வேனியா யூனிவர்சிடி அதிகாரிகளுக்குச் சமர்ப்பித்தார்; அது காணுமல் போய்விட்டது. மீண்டும் அதை மாச்லே தமது குறிப்புகளிலிருந்து எழுதிச் சமர்ப்பிக்க வேண்டிய நிலை ஏற்பட்டது. இந்தக் கம்ப்யூட்டர் (ENIAC) ஒரு தனிப்பட்ட (special purpose) கம்ப்யூட்டர் ஆகும்; பீரங்கிக் குண்டுகள், விமானங்கள், கப்பல்கள் ஆகியவற்றிலிருந்து வீசப்படும் குண்டுகள் ஆகியவைகளின் பாதை, வேகம் முதலியவற்றைக் கணிப்பதற்கென்றே இக் கம்ப்யூட்டர் தயாரிக்கப்பட்டது. இக் கம்ப்யூட்டரில் 500,000 இணைப்புகளும் (Soldering), 18,000 வால்வுகளும், 6,000 ஸ்விட்ச்களும், 5,000 மின் முனைகளும் (Terminals) இருந்தன. பாச்லே, எக்கர்ட் உட்பட, 12 பொறியியலாளர் இரண்டரை ஆண்டுகள் உழைத்து இதைச் செய்து முடித்தனர்.

கணிப்பு வேகத்தை ஒப்பிடும்போது மார்க்-I ஏனியாக்கை விட மிகவும் மெதுவானது. மார்க்-I ஒரு நொடியில் 3-எண்களைக் கூட்டும் அளவுக்கு மட்டுமே வேகம் பெற்றிருந்தது. ஆனுல் ஏனியாக் அதே நேரத்தில் 5,000 கூட்டல்களைச் செய்தது. இந்தக் கம்ப்யூட்டரில் பற்சக்கரங்கள் (gears) முற்றிலுமாக நீக்கப்பட்டுவிட்டன. மின்துடிப்புகளைக் கொண்டு (electronic pulses) எண்ணும் முறை (counting method) கையாளப்பட்டன; கணிப்பின் விடைகள் துளைக் கார்டுகள் வழியாகப் பெறப்பட்டன. புரோகிராம் ஆணைகள், கம்ப்யூட்டரின் உள்ளாக வைக்கப்பட்டுச் செயல்படுத்தப்பட்டன. ஆனுல், இவ்வாணைகளைக் கம்ப்யூட்டரில் இட, நூற்றுக் கணக்கான மின் இணைப்புகளைச் செய்ய வேண்டியிருந்தது. இதற்கு மட்டுமே பல நாள்கள் ஆயின. இந்தத் தொல்லைகளிலிருந்து விடுபடுவதற்கென்று வேறு ஒரு முறை வந்தது. அதுவே கம்ப்யூட்டர் நினைவகத்தில் புரோகிராம் ஆணைகளைச் சேர்த்துவைக்கும் முறை (Stored Programme Method) யாகும். இம் முறையை முதன் முதலாகக் கண்டுபிடித்தவர் யார்

என்பது தெளிவாகத் தெரியவில்லை. எனினும் 1946ஆம் ஆண்டு 'பர்க்ஸ்', 'கோல்ட்ஸ்டீன்', 'வான்நாய்மன்' ஆகிய மூவரும் சேர்ந்து எழுதிய ஓர் ஆராய்ச்சிக் கட்டுரையில் இம்முறை விரிவாகச் சித்திரிக்கப்பட்டுள்ளது.

கம்ப்யூட்டர் ஆராய்ச்சியில் பெருமளவுக்கு ஈடுபாடு கொண்டிருந்தது அமெரிக்காவென்ருலும், புரோகிராமை நினைவகத்திட்டுச் செயல்படும் (Stored Programme) கம்ப்யூட்டரை முதலில் செய்தது இங்கிலாந்து ஆகும். எட்சாக் (EDSAC) என்று சுருக்கமாக அழைக்கப்பட்ட இக் கம்ப்யூட்டர் கேம்பிரிட்ஜ் பல்கலைக் கழகத்தில் 1949ஆம் ஆண்டு செய்து முடிக்கப்பட்டது. இதைத் தொடர்ந்து அமெரிக்காவில் மாச்லேவும், எக்கர்ட்டும் எட்சாக்கைப் போன்ற புதிய கம்ப்யூட்டர் ஒன்றைப் பென்சில்வேனியா பல்கலைக்கழகத்தில் வடிவமைத்துச் செய்ய முற்பட்டார்கள். இந்தக் கம்ப்யூட்டர் முடிவு பெறுவதற்கு முன்னரே இவ்விருவரும் பல்கலைக் கழகத்தை விட்டுப் போய் கம்ப்யூட்டர் உற்பத்தி செய்யும் கம்பெனி ஒன்றை ஆரம்பித்தார்கள். இந்தக் கம்பெனி பின்னர் ரெமிங்டன்—ரேண்ட் துறையினரால் (Remington-Rand Corporation) வாங்கிக் கொள்ளப்பட்டது. அதன் பின்னர் எக்கார்ட் ரேண்ட் கார்ப்பரேஷனிலேயே தங்கிவிட்டார். ஆனால் மாச்லே மீண்டும் பிரிந்து போய் மாச்லே அசோசியேட்ஸ் இன்கார்ப்பரேட்டட்' (Mauchley Associates Inc.) என்ற வேருெரு கம்பெனியை ஆரம்பித்துக்கொண்டார்.

பர்க்ஸ், கோல்ட்ஸ்டீன், வான்நாய்மன் ஆகியோர் ஒன்று சேர்ந்து எழுதிய ஆராய்ச்சிக் கட்டுரையில் விவரிக்கப்பட்டிருந்த பல வடிவமைப்பு விவரங்கள், அன்றைய கணிப் பொறிகளின் அமைப்பில் பல அடிப்படை மாறுதல்களை வலியுறுத்தின. வான்நாய்மனின் கம்ப்யூட்டருக்கும் இன்று பயன்படுத்தப்பட்டுக் கொண்டிருக்கும் கம்ப்யூட்டர்களுக்கும் மிகச் சில வேறுபாடுகளே உள்ளன. இதிலிருந்தே நாய்மன் கம்ப்யூட்டரின் வடிவமைப்பின் திண்மை புலப்படும். இரண்டாம் உலகப் போருக்குப் பின்னர் நாய்மன் எடுத்துக் கொண்ட முதல் வேலை, தான் விவரித்திருந்த கம்ப்யூட்டரைச் செய்து முடிப்பதே. பிரின்ஸ்டன் உயர்நிலைக் கல்விக்கழகத்தில் (Institute of Advanced Study at Princeton) தமது பதவிக்குத் திரும்பிப் போய்ச் சேர்ந்ததும், நாய்மன் தமது வேலையை ஆரம்பித்தார். நாய்மனின் கம்ப்யூட்டருக்கான பொருளுதவியை அமெரிக்கத் தரைப்படைப் பிரிவு செய்தது. எனினும் கம்ப்யூட்டர் செய்து முடிக்கப்பட்டவுடன் அவர்கள் அதை எடுத்

துச்செல்லாமல் பிரின்ஸ்டன் கழகமே வைத்துக்கொள்ள அனுமதித்தனர். இதையே பிற்காலத்தில் 'IAS மெஷின்,' 'பிரின்ஸ்டன் மெஷின்,' 'நாய்மன் மெஷின்' என்று பலவிதமாக அழைத்தனர்.

நாய்மன் கம்ப்யூட்டரைத் தொடர்ந்து பலவிதமான கம்ப்யூட்டர்கள் உற்பத்தி செய்யப்பட்டன. ஆர்கான் நேஷனல் சோதனைச் சாலையைச் சேர்ந்த ஆவிடாக் (Avidac), ஓகரிட்ஜைச் சேர்ந்த ஒரேக்ல் (Oracle), இல்லினுய்ஸ் பல்கலைக் கழகத்தைச் சேர்ந்த இல்லியாக் (Illiac), ஆபர்டீனைச் சேர்ந்த ஆர்ட்வாக் (Ordvac), லாஸ் ஆல்மாவைச் சேர்ந்த மேனியாக்-I (Maniac-I), சிட்னி பல்கலைக் கழகத்தைச் சேர்ந்த சில்லியாக் (Silliac), ரேண்டு கார்ப்பரேஷன் தயாரித்த ஜான்னியாக் (Johnniac) (இப்பெயர் ஜான்வான் நாய்மனின் முதல் பெயரைக் கொண்டு உருவாக்கப்பட்டது), ஐபிஎம் (IBM) 701 ஆகியவை இவற்றுள் சில.

நாய்மன் கம்ப்யூட்டரின் அடிப்படையில் உருவாக்கப்பட்ட பல வேறு கம்ப்யூட்டர்களும், சிறு சிறு தனிச் சிறப்புகளைக் கொண்டிருந்தன. எடுத்துக்காட்டாக மேனியாக் 1024 'வார்த்தைகள்' கொள்ளும் நினைவகத்தைக் கொண்டிருந்தது. (இன்றைய கம்ப்யூட்டர்களின் நினைவகத்தில் மில்லியன் கணக்கான வார்த்தைகளை இருத்த முடியும்.) 80 வார்த்தைகள் நினைவகத்தையே பெரும்பாலும் கொண்டிருந்த அக்காலக் கம்ப்யூட்டர்களை ஒப்பு நோக்கும்போது 1024 வார்த்தை கம்ப்யூட்டர் ஒரு மாபெரும் பொறியாகத் தோன்றியிருக்கக்கூடும். இந்த மானியாக்தான் அமெரிக்கா, மற்ற எல்லா நாடுகளையும் முந்திக் கொண்டு ஹைட்ரஜன் குண்டைத் தயாரிக்கக் காரணமாயிருந்தது என்று கூறப்படுகிறது.

நாய்மனின் கம்ப்யூட்டருக்கும், இன்றைய கம்ப்யூட்டருக்கும் இருக்கும் வேறுபாடுகள் எல்லாம், கணிப்பு வேகம் நினைவக அமைப்பு, தொடர்பு முறை (Communication Method) ஆகியவைபற்றியே தவிர அடிப்படைக் கருத்தில் ஏதுமில்லை. இன்று நாம் கையாளும் கம்ப்யூட்டர்களின் நேரடி முன்னோடி, நாய்மன் கம்ப்யூட்டராம்.

அடுத்து வரும் பக்கங்களில் இத்தகைய கம்ப்யூட்டர்களும், அவற்றிற்கு உதவியாக உள்ள மற்ற உதவிப் பொறிகளும் (Peripheral biniss) எவ்வாறு இயங்குகின்றன என்பது பற்றியும் பார்ப்போம்.

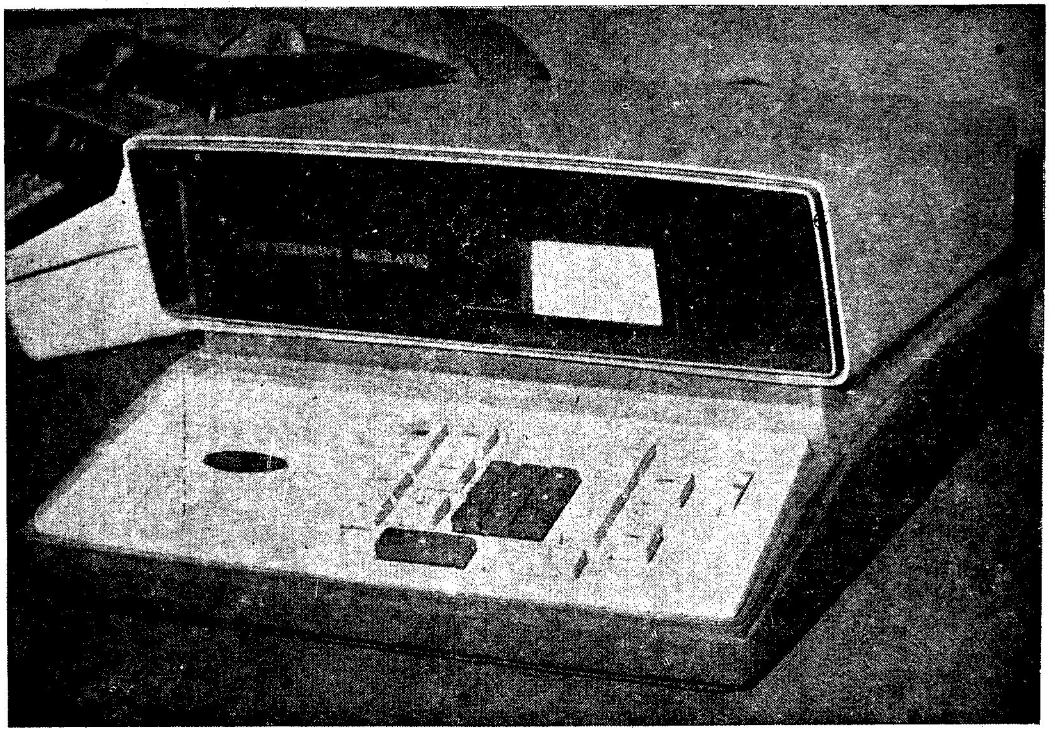
1

1. கம்ப்யூட்டர் என்பது என்ன ?

கம்ப்யூட்டர் என்ற சொல்லுக்கு நேரான பொருள் தேடினுல் 'கணிக்க உதவும் கருவி' என்று விளக்கம் கிடைக்கும். கணிப்பதற்கு (அது எதுபற்றிய கணிப்பாயினும்), கருவியாகப் பயன் படுவது கம்ப்யூட்டர் ஆகும். கையிலுள்ள பத்து விரல்களால் நம் வீட்டுச் சிறுவன் பள்ளிக்கணக்குகளைச் செய்கிருன்; வாடகைக் காரில் உள்ள மீட்டர் காரோட்டிக்கு நாம் செலுத்த வேண்டிய தொகையைக் கணிக்கிறது; சுவரில் உள்ள கடிகாரம் கழிந்து செல்லும் காலத்தைக் கணிக்கிறது; இவை யாவும் கம்ப்யூட்டர்களே. இத்தகைய கம்ப்யூட்டர்களை நம்மைச்சுற்றி எங்கெங்கும் காணலாம். பார்க்கப்போனுல் இத்தகைய எளிய கம்ப்யூட்டர்கள், வீட்டிலும் வெளியிலும் நம் அன்ருட வாழ்வில் இன்றியமையாதவையாய் அமைந்துவிட்டன.

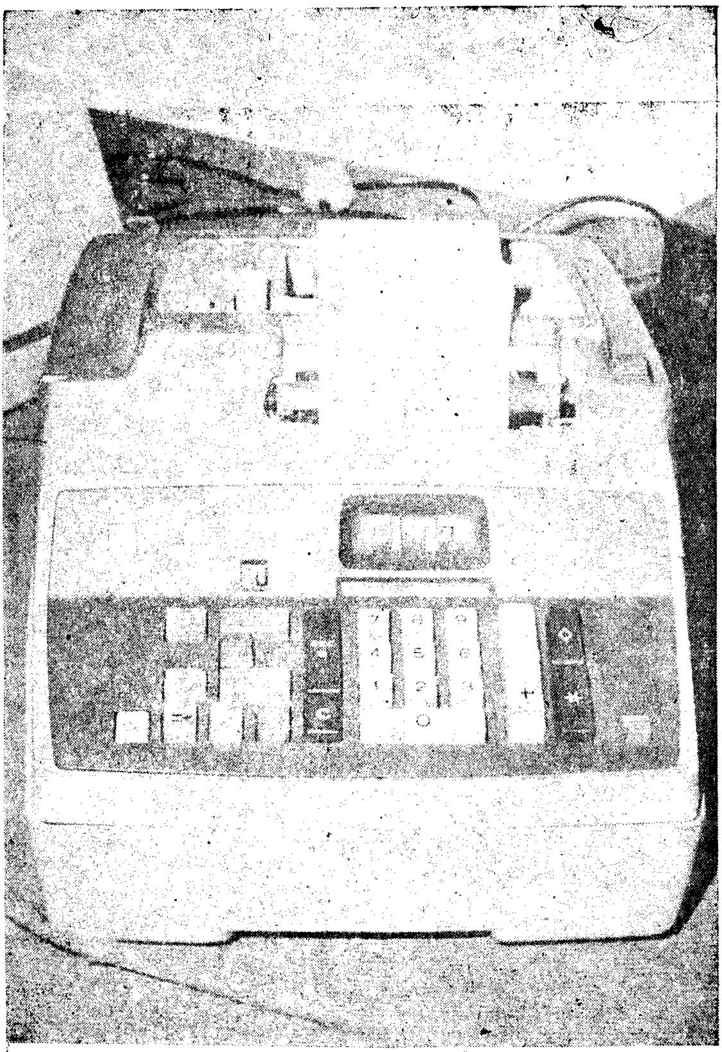
மேற்கூறியவையன்றி கூட்டல், கழித்தல், பெருக்கல், வகுத்தல் போன்ற கணக்குகளை — எண்களைக் கையாண்டு — செய்ய வல்ல கம்ப்யூட்டர்களையும் நாம் பார்க்கிருேம். ஸ்லைட் ரூல், டெஸ்க் கால்குலேடர் முதலியவை இவ்வினத்தைச் சாரும். இக் கம்ப்யூட்டர்களுக்கும் முன் சொன்ன கம்ப்யூட்டர்களுக்கும் அடிப்படை வேறுபாடுகள் உண்டு. முன்சொன்ன கருவிகள் ஒவ்வொன்றும் ஒவ்வொரு குறிப்பிட்ட இயல்புள்ள பொருளையே அளக்கக் கூடியவை. எடுத்துக்காட்டாக கடிகாரம் காலத்தை மட்டுமே கணக்கிடும்; தெர்மாமீட்டர் வெப்பத்தின் அளவை மட்டுமே காட்டும். ஆனுல் ஸ்லைட்ரூல், டெஸ்க் கால்குலேடர் ஆகியவற்றைக் கொண்டு நாம் பலவிதக் கணக்குகளையும், வெறும் எண்களைக் கொண்டு செய்ய இயலும்.

இப் புத்தகத்தில் நாம் கற்கப்போவது இந்த இருவகை கம்ப்யூட்டர்களைப் பற்றியும் அல்ல. நாம் கற்கப்போகும் கம்ப்யூட்டர்கள் மிகவும் சிக்கலான வடிவமைப்பு (design) கொண்



படம் *1.1* எலக்ட்ரானிக் டெஸ்க் கால்குலேட்டர்

இந்த கணிப் பொறி முழுவதும் ட்ரான்ஸிஸ்டர்களால் ஆனது. 'இதன் கணிப்பு வேகம்' கம்ப்யூட்டரின் கணிப்பு வேகத்திற்கு இணையானது. கணிப்பில் ஏற்படும் கூட்டல், கழித்தல் ஆகிய செயல்களுக்கு உரிய பித்தான்களை அழுத்தி அந்தந்தக் கணிப்பைச் செய்யவேண்டும். ஒவ்வொரு கணிப்பு முடிந்ததும், கணிப்பின் விடை, பித்தான்களுக்கு மேலே காணப்படும் பலகணியில் தோன்றும். அவற்றை (வேண்டுமாயின்) நாம் [illegible] எழுதிக்கொள்ள வேண்டும்.



படம் 1.2

எலக்ட்ரானிக் டெஸ்க் கால்குலேட்டர்

படம் 1.1ல் உள்ளதைப் போன்றதே, படம் 1.2ல் உள்ள கால்குலேட்டரும். படம் 1.2ல் உள்ள கால்குலேட்டரில் கணிப்பினூடே கிடைக்கும் முடிவுகள் (Results) அதில் இணைக்கப்பட்டுள்ள காகிதத்தில் பொறிக்கப்படுகின்றன.

டவை; விலையுயர்ந்தவை; குளிர்வறைகளில் (**Air Conditioned Rooms**) இயங்குபவை. இவற்றை இயக்குவோரும், பயன்படுத்த உதவுவோரும் தனிப்பயிற்சி பெற்றவர்கள். இவ்வியந்திரங்களைக் 'கம்ப்யூட்டர்' என்ற ஒற்றைச் சொல்லால் மட்டும் குறிப்பிட்டால் போதாது. இவற்றின் இயல்பு ஓரளவு வெளிப்பட 'கட்டளைக் கொத்தேற்று, தன்னியக்கும், பெருவேக எலக்ட்ரானிக் கம்ப்யூட்டர்' (**Stored program automatic, high speed electronic computor**) என்று அழைப்பதே பொருத்தம். இந்த நெடிய பெயரில் உள்ள பல்வேறு அடைமொழிகளை ஆய்வதிலேயே இன்றைய கம்ப்யூட்டர்களின் இயல்புகளையும், ஆற்றல்களையும் நாம் ஓரளவு அறியக்கூடும். பின்வரும் பக்கங்களில் இந்த அடைமொழிகள் ஒவ்வொன்றையும் சற்றுப் பார்ப்போம்.

மிகை வேகம் (High Speed)

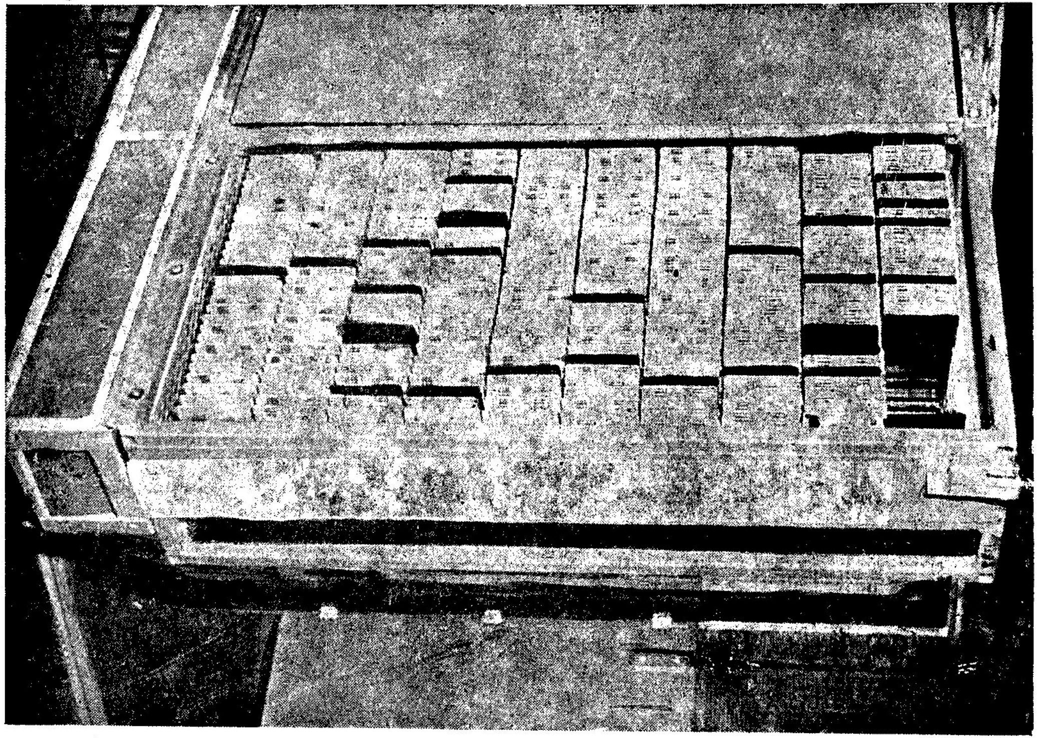
நவீன கம்ப்யூட்டர்கள் வேகத்திற்குப் பெயர்போனவை. ஒரு நிமிடத்திற்கு, ஒவ்வொன்றும் 10 இலக்கங்களை உடைய ஒரு மில்லியன் எண்களைக் கூட்டவல்ல கம்ப்யூட்டர்கள் இன்று உள்ளன. நூறு மனிதர்கள் சேர்ந்து நூறாண்டுக் காலத்தில் செய்யும் கணக்கை, ஓரிருமணி நேரத்தில் இக் கம்ப்யூட்டர்கள் செய்கின்றன. மனிதர்களின் கணிப்பில் பிழைகள் இருக்கக் கூடும். ஆனால், கம்ப்யூட்டரின் கணிப்போ துல்லியமானது. சாதாரண மனிதனின் கற்பனைக்கு எட்டாத அளப்பறிய வேகமும், துல்லியமும் கம்ப்யூட்டரின் இருபெரும் சிறப்பியல்புகளாகும். இவ்விரு இயல்புகளும் இன்று பல்வேறு விஞ்ஞானத் துறைகளிலும் வேண்டப்படுகின்றன. எடுத்துக்காட்டாக விண்வெளி ஆராய்ச்சி, விண்வெளிப் பயணம் ஆகியவற்றைக் கூறலாம். கம்ப்யூட்டர்கள் மட்டும் இல்லையாயின் விண்வெளிப் பயணமும், ஆய்வும் ஆரம்பித்தே இருக்கமுடியாது. ஒவ்வொரு பயணத்திற்கு முன்னரும், நூற்றுக்கணக்கான நிகழ்வுப் பயணங்கள் (**Simulated flights**) ஆய்வுக்கூடங்களில் மேற்கொள்ளப் படுகின்றன. இப் போலிப் பயணங்கள் மூலமாக, உண்மைப் பயணத்தின்போது எதிர்பாராமல் ஏற்படக்கூடிய தவறுகள், விபத்துகள் ஆகியவை பற்றியும், அப்படி அவை ஏற்பட்டால் அவற்றைச் சரிசெய்ய அல்லது சமாளிக்கவேண்டிய வழிமுறைகள்பற்றியும் கற்றுத் தெளியப்படுகின்றன. அதனால் பின்னர் உண்மையான பயணம் நிகழும்போது பயணத்தை மேற்கொள்ளுவோரும், அதைக் கண்காணிக்கும் தரைக்கட்டுப்பாட்டு நிலையத்தோரும் எத்தகைய விளைவுகளுக்கும் ஆயத்தமாயிருக்கிறார்கள். கம்ப்யூட்டர்கள் இல்லையாயின் போலிப்பயண ஆய்வு இயலாது.

அதுவுமன்றி, பின்னர் உண்மையான பயணம் நிகழும்போது ஒவ்வொரு நொடியும், பயணப்பாதை, வேகம், பயணிகளின் உடல்நிலை, மனநிலைகளின் பல்வேறு அம்சங்கள் யாவற்றையும் கண்காணித்து உடனுக்குடன் தேவையான மாறுதல் அல்லது சீரமைப்பு செய்தாகவேண்டும். கம்ப்யூட்டரின் அளவற்ற வேகம் இதைச் சாத்தியமாக்குகிறது.

விண்வெளிப்பயணம் மட்டுமன்றி இன்னும் பல்வேறு துறைகளிலும் கம்ப்யூட்டர் இன்றியமையாததாய் உள்ளது. பாதுகாப்பு, மருத்துவம், கல்வி, விஞ்ஞானம், பொறியியல் ஆகியவற்றிலும் இன்னும் மற்றும் பல்வேறு துறைகளிலும் பெருமளவில் கம்ப்யூட்டர்கள் இன்று பயன்படுத்தப்படுகின்றன. இவற்றைப்பற்றி விரிவாகப் பின்னர் காண்போம்.

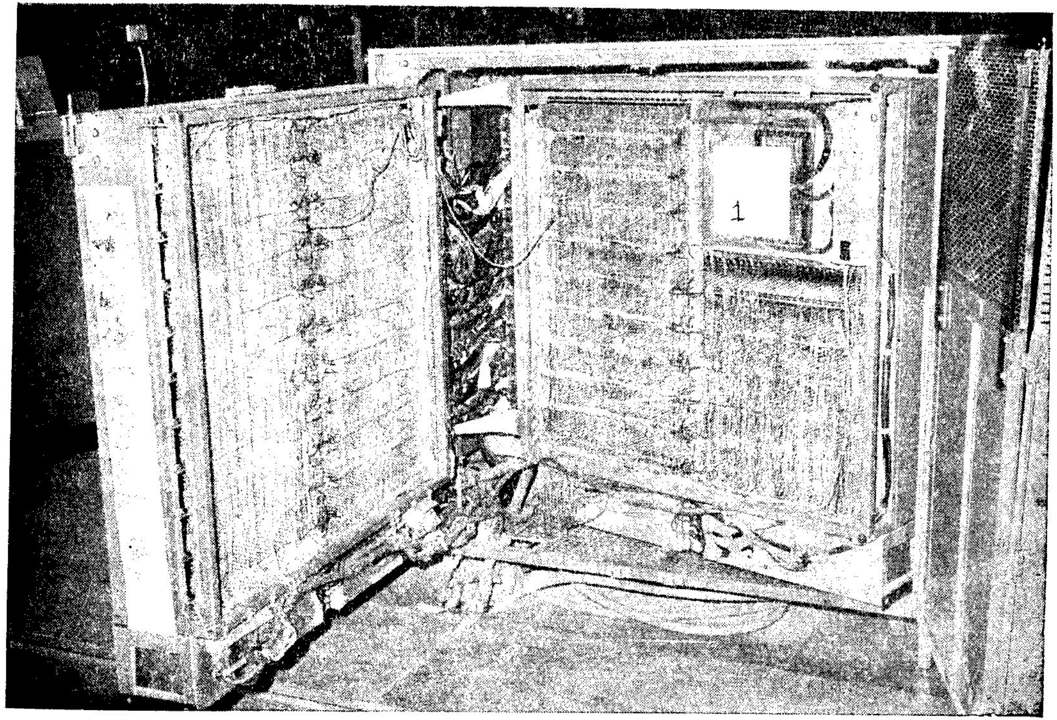
எலக்ட்ரானிக் (Electronic)

கம்ப்யூட்டர்களின் கற்பனைக்கரிய வேகத்திற்குக் காரணம், அதன் முக்கிய பாகங்கள் யாவும் எலக்ட்ரானிக் கருவிகளால் (Electronic Components) ஆக்கப்பட்டிருப்பதே. பார்க்கப் போனால் இன்றைய கம்ப்யூட்டரின் வளர்ச்சி முழுவதும் எலக்ட்ரானிக் துறையில் ஏற்பட்ட மலர்ச்சியின் விளைவேயாகும். இத்துறையில் ஏற்பட்டுள்ள வியத்தகு வளர்ச்சியை முற்றும் பயன்படுத்திக் கொண்டதன் மூலம்தான் கம்ப்யூட்டர் துறையில் அளப்பரிய முன்னேற்றங்கள் ஏற்பட்டுள்ளன. ஆரம்ப காலத்தில் கம்ப்யூட்டர்கள் முற்றும் முடுக்கிகளாலும் (gears) நெம்பிகளாலும் (levers) நிரப்பப்பட்டிருந்தன. பின்னர் வந்த கம்ப்யூட்டர்கள், வெற்றுமின்குழல் (Vacuum tubes) களைக் கொண்டு இயங்கின. எலக்ட்ரானிக் துறையில் டிரான்ஸிஸ்டர்கள் கண்டுபிடிக்கப்பட்ட பின்னர், வெற்றுக்குழல்களுக்கு மாறாக டிரான்ஸிஸ்டர்கள் பயன்படுத்தப்பட்டன. இதன் விளைவாக கம்ப்யூட்டரின் பருமன் மிகவும் சுருங்கியதோடு செயல்வேகமும் அதிகரித்தது. டிரான்ஸிஸ்டர்களைக் காட்டிலும் அளவில் மிகச் சுருங்கிய ஒன்றிய மின்வழிகள் (Integrated Circuits) கண்டுபிடிக்கப்பட்ட பின்னர் எலக்ட்ரானிக் துறையில் நுண்-சுருக்க முறை (Micro Miniaturisation) தீவிர வளர்ச்சியடைந்தது. சில மின்வழிகளில் (Circuits) உள்ள குறைகடத்திகள் (Semi-Conductors) *0.005"* குறுக்கும், *0.06"* நெடுக்கும் உள்ளவை; இன்றைய நுண்சுருக்க முறையின் தீவிரத்தை இது ஓரளவு புலப்படுத்தும். இன்று டிரான்ஸிஸ்டருக்குப் பதிலாக ஒன்றிய மின்வழிகள் (Integrated Circuits) கொண்ட கம்ப்யூட்டர்கள் வடிவமைக்கப்பட்டு உற்பத்தி செய்யப்படுகின்றன. இவ்வாறாக, இன்றைய கம்ப்யூட்டர்கள் தமது தற்போதைய



படம் 1.2a

IBM 1620 கம்ப்யூட்டரின் ட்ரான்ஸிஸ்டர் அட்டைகளில் சில



படம் 1.2b

IBM 1620 கம்ப்யூட்டரின் உள்தோற்றம். படத்தில் 1 என்று குறிக்கப்பட்டுள்ளது கம்ப்யூட்டரின் நினைவகம்

உன்னத நிலைமைக்கு, எலக்ட்ரானிக் துறைக்கு பெருமளவில் கடமைப்பட்டுள்ளன.

நினைவகக் கட்டளைக் கொத்தேற்று... (Stored Program)

கம்ப்யூட்டர்கள் பெருவேகம் வாய்ந்தவை என்றும், அவற்றின் அதிவேகத்திற்குக் காரணமாயிருப்பவை, மூலபாகங்களான எல்க்ட்ரானிக் சாதனங்கள் என்றும் அறிந்தோம். இப்போது 'கட்டளைக் கொத்து' (அல்லது புரோக்ராம் Program) என்றால் என்ன, அதை ஏற்றுச் செயல் படுவது எப்படி என்பனவற்றை ஒரு சிறு எடுத்துக்காட்டுடன் பார்ப்போம்.

எனக்கு ஒரு சிறு பிரச்சினை; அருகிலுள்ள கடைக்குப் போய் ஒரு புத்தகத்தை வாங்கி வர வேண்டும். நானே கடைக்குப் போய் அப்புத்தகத்தை வாங்கி வர முடியும். ஆனால் நான் அதைச் செய்ய விரும்பவில்லை. மாறாக வீட்டில் உள்ள சிறுவன் ஒருவனை அனுப்பி அப்புத்தகத்தைப் பெற விரும்புகிறேன். ஆனால் சிறுவனுக்கோ கடைக்கு வழி தெரியாது. அதனால் கீழ்கண்டவாறு அவனுக்கு நான் புத்தகம் வாங்கி வரும் வழிமுறைகளைக் கூறுகிறேன்.

1. வீட்டு வாசலுக்குப் போய்த் தெருவில் இறங்கு.
2. உன் இடதுபுறமாக 200 மீட்டர் தொலைவு நடந்து செல்.
3. தெருவைக் குறுக்காக கடந்து கொள்.
4. கடந்தவுடன் உனது இடது புறம் மூன்றாம் நுழைவாயிலுக்குச் (புத்தகக்கடை) செல்.
5. புத்தகத்தை வாங்கிக் கொள்.
6. சென்ற வழியே திரும்பிவா.

மேற்கண்ட ஆறு கட்டளைகளையும் சிறுவன் கவனமாகக் கேட்டு, நினைவில் கொண்டு, வழுவாது, வரிசை பிறழாது ஒன்றன் பின் ஒன்றாகச் செய்து முடிப்பானாயின் என்னுடைய பிரச்சினை தீர்வு பெறும். அதாவது நான் வாங்க விரும்பிய புத்தகம் எனக்கு வந்து சேரும்.

சற்று சிந்தித்துப் பார்த்தால், இந்த சிறு பிரச்சினை மட்டும் அல்லாமல், எந்தப் பிரச்சினைக்கும்—சிறியதாயினும் பெரியதாயினும்—அதைத் தீர்ப்பதற்கான வழிமுறையை, சிறு சிறு

எளிய, தர்க்க ரீதியாக அமைந்த, கட்டளைகள் வடிவில் கொடுக்க இயலும் என்பது தெரிய வரும். இச்சிறு எளிய கட்டளைகளை அவை கொடுக்கப்பட்டுள்ள ஒழுங்கில் வழுவாது, பிசகாது செய்து முடித்தால் அப்பிரச்சினைக்குத் தீர்வு காண லாம்.

விளக்கம் (Definition)

ஒரு பிரச்சினையைத் தீர்ப்பதற்கென, தர்க்க ரீதியில் அமைந்த எளிய சிறிய கட்டளைகளின் தொகுப்பே 'கட்டளைக் கொத்து' அல்லது புரோக்ராம் (Program) எனப்படும்.

முன் கூறிய ஆறு கட்டளைகளும் புத்தகம் வாங்கிவரும் பிரச்சினையைத் தீர்ப்பதற்கான புரோக்ராம் (கட்டளைக் கொத்து) ஆகும். இந்த எடுத்துக்காட்டில் நமது சிறுவன் தனக்கிடப் பட்ட ஆறு கட்டளைகளையும் நிறைவேற்ற வேண்டுமாயின் முதலாவதாக அவன் அக்கட்டளைகளை நம்மிடமிருந்து அவனது மூளையில் ஏற்று வரிசைக்கிரமமாக நினைவில் வைத்துக் கொள்ளல் இன்றியமையாதது. இவ்விதம் கட்டளைகளை **நினைவில் ஏற்கும் ஆற்றலையே** நாம் 'கட்டளைக் கொத்தேற்பது' என்று குறிப்பிடு கிறோம்.

மேற் சொன்ன சிறுவனுக்கு அவனுக்கிடப்பட்ட கட்டளைகளை நினைவில் வைத்துக் கொள்ளும் ஆற்றல் உள்ளது. அதே போல் இன்றைய கம்ப்யூட்டர்களும் நாமிடும் கட்டளைகளை ஏற்று, நினைவில் கொண்டு, பின்னர் அவற்றை ஒன்றன் பின் ஒன்றாக செய்து முடிக்க வல்லன. நாம் செய்ய விரும்பும் கணிப்பு எத்துறையைச் சேர்ந்ததாயினும்— விஞ்ஞானம், பொறியியல் கல்வி, சமூகவியல், பொருளாதாரம், கலை, விவசாயம் முதலிய எத்துறையைச் சார்ந்திருப்பினும்— அக்கணிப்பிற்கான வழி முறைகளை கம்ப்யூட்டரின் ஆற்றலுக்கு உட்பட்ட எளிய, சிறிய ஆணைகள் வடிவில், ஒரு புரோக்ராம் மூலம் கொடுப்போமாயின் இவ்வாணைகளை நினைவில் கொண்டு அவற்றை நிறைவேற்ற கம்ப்யூட்டர்கள் காத்திருக்கின்றன.

சற்று முன் சொன்ன எடுத்துக் காட்டில் 'கடைக்குப் போய் புத்தகம் வாங்கி வா' என்ற கட்டளையை நம் சிறுவன் நிறை வேற்ற இயலாது. (ஏனெனில் கடைக்குச் செல்லும் வழி அவனுக்குத் தெரியாது.) அது அவனது ஆற்றலுக்கு உட்பட்ட தில்லை. ஆனால் அதே சமயத்தில் நாம் முன்னர் கொடுத்த ஆறு கட்டளைகளில் ஒவ்வொன்றும் அவனது ஆற்றலுக்கு, உட்பட்டிருக்கின்றது. இந்த ஆறு கட்டளைகளின் தொகுதி

மேற்சொன்ன ''கடைக்குப் போய் புத்தகம் வாங்கி வா'' என்ற ஒரு பெரும் கட்டளைக்கு ஈடு ஆகும். இது போலவே நாம், கணிப்பின் செயல் முறைகளைக் கட்டளைகளாக மாற்றிக் கம்ப்யூட்டருக்கான கட்டளைக் கொத்து ஒன்றைத் தயாரிக்கும் போது அக்கட்டளைகள் ஒவ்வொன்றும் கம்ப்யூட்டரின் செயலாற்றலுக்கு உட்பட்டவையாயிருத்தல் அவசியம். நாம் இத்தகைய கட்டளைகளைத் தயாரிக்க கம்ப்யூட்டர் என்னென்ன செய்யவல்லது என்ற அறிவு நமக்குத் தேவையாகிறது.

தற்காலக் கம்ப்யூட்டர்கள் கூட்டல், கழித்தல், பெருக்கல், வகுத்தல் என்ற நான்கு அடிப்படைக் கணிப்புகளுடன், ஒப்பு நோக்கும் ஆற்றலும் பெற்றுள்ளன. நாம் ஏதாவது இரண்டு எண்களைக் கொடுத்தால் அவையிரண்டும் சமமா இல்லையா, இல்லையாயின் அவற்றுள் எது பெரியது என்றும் கம்ப்யூட்டர் காணவல்லது. அதே போல் இரண்டு குறியீடுகளைக் கொடுப்பினும் அவை மாறுபட்டவையா, இல்லை அவையிரண்டும் ஒரே குறியீடுதானு என்றும் காணவல்லது. எடுத்துக்காட்டாக A, B என்ற இரு எழுத்துக்களை வெவ்வேறு குறியீடுகள் என்ற கம்ப்யூட்டர் அடையாளம் காணவல்லது. அதே சமயத்தில் இரண்டு முற்றிலும் ஒத்தக் (Identical) குறியீடுகளைக் கொடுத்தால் அவையிரண்டும் ஒன்றே என்றும் அறிய வல்லது. எடுத்துக்காட்டாக இரண்டு A என்ற எழுத்துக்களைக் கொடுத்து ஒப்ப நோக்கச் சொன்னுல் அவை இரண்டும் ஒன்றே என்று கம்ப்யூட்டர் 'அறிய' வல்லது.

அட்டவணை 1.1ல் IBM 1401 என்ற கம்ப்யூட்டரின் செயலாற்றல்கள் கொடுக்கப் பட்டுள்ளன.

நாம் தயாரிக்கும் கட்டளைக் கொத்தில் உள்ள கட்டளை ஒவ்வொன்றும், நமது கம்ப்யூட்டரின் செயலாற்றல்களுக்கு உட்பட்ட ஆணையாக முறிக்கப் பட்டிருக்க வேண்டும் என்று முன்னர் பார்த்தோம். எடுத்துக் காட்டாக நாம் IBM 1401 என்ற கம்ப்யூட்டரை உபயோகப் படுத்தி ஏதாவது ஒரு கணிப்பு செய்ய வேண்டுமாயின், அக்கணிப்பின் வழிமுறைகளைக் கொண்ட கட்டளைக் கொத்தின் ஆணை ஒவ்வொன்றும் அட்டவணை 1.1ல் உள்ள ஏதாவது ஒரு ஆணையாகத்தான் இருக்கவேண்டும்.

தன்னியக்கம்

கட்டளைக் கொத்தேற்கும் ஆற்றலுடன், இன்றைய கம்ப்யூட்டர் தன்னியக்கத் தன்மையும் (automatic) கொண்டது.

வரிசை எண்	செயல் குறியீடு	விளக்கம்	
1	,	Set word mark	களக் குறியீடு இடு
2	⌣	Clear word mark	களக் குறியீட்டை அழி
3	M	move	கொண்டு செல்
4	L	Load	ஏற்று
5	Z	Move and zero suppress	கொண்டுசென்று பூச்சியங்களை எடு
6	D	move digit	எண்ணை எடுத்துச் செல்
7	Y	move zone	தொகுதிக் குறியீட்டை எடுத்துச் செல்
8	1	Read	படி
9	4	Punch	துளை
10	2	Print	அச்சிடு
11	A	Add	கூட்டு
12	S	subtract	கழி
13	?	Zero and add	பூச்சியமாக்கி அத்துடன் கூட்டு
14	!	Zero and subtract	பூச்சியமாக்கி அதிலிருந்து கழி
15	C	Compare	ஒப்பு நோக்கு
16	B	Branch	பிரிந்து செல்
17	V	Test for Zone or W/M	தொகுதிக் குறி அல்லது களக்குறி இருக்கிறதா என்று பார்
18	/	Clear storage	நினைவகத்தைச் சுத்தி செய்
19	N	no operation	செயல் ஏதும் இல்லை
20	.	Stop	நில்
21	K	stacker select	படிப்பொறி சேர்ப்பறையைத் தேர்ந்துகொள்

அட்டவணை 1·1

ஐ. பி. எம். 1401 கம்ப்யூட்டரின் ஆணைகள்
(மேலுள்ள அட்டவணை முற்றுப் பெற்றதல்ல)

தன்னியக்கத் தன்மையென்றால் என்ன ? சற்று முன் சொன்ன எடுத்துக்காட்டிற்கு மீண்டும் செல்வோம். நம் சிறுவன் செயல்படும் முறையில், முதல்படி கட்டளைகளை ஏற்றுக்கொண்டு அவற்றை நினைவு கொள்வதாகும். அடுத்தபடி அக்கட்டளைகளை ஒவ்வொன்றாக—முதல் கட்டளையில் ஆரம்பித்து—நிறைவேற்றி முடிப்பதாகும். நம் சிறுவன் ஒரு கட்டளையை செய்து முடித்த பின் அடுத்த கட்டளையைத் **தானாகவே** மேற்கொண்டு செய்ய ஆரம்பிக்கிறான். அதுவும் முடிந்த பின் மீண்டும் **அவனாகவே** அதற்கடுத்த கட்டளையை மேற்கொண்டு நிறைவேற்றுகிறான். எடுத்துக் காட்டாக முதல் கட்டளையை எடுத்துக் கொள்ளுவோம். இது 'வீட்டு வாசலுக்குப் போய்த் தெருவில் இறங்கு'என்பதாகும். இவ் வாணையை நிறைவேற்றிய பிறகு, (அதாவது தெருவில் இறங்கிய பிறகு)— தன் நினைவிலிருந்து **தானாகவே** அடுத்த கட்டளையை ('உன் இடது புறமாக 200 மீட்டர் தொலைவு நடந்து செல்') மேற் கொண்டு அதனை நிறைவேற்றும் பொருட்டு இடதுபுறம் திரும்பி நடக்க ஆரம்பிக்கின்றான். இதை முடித்த பிறகு மீண்டும் **தானாகவே** மூன்றாவது ஆணையை ('தெருவைக் குறுக்காக கடந்து கொள்') நினைவு கூர்ந்து தெருவைக் கடக்கிறான். இதில் குறிப்பிடத்தக்கது யாதெனில் ஒவ்வொரு ஆணையையும் நிறைவேற்றிய பிறகு **நமது தலையீடு இல்லாமல்** தானாகவே அடுத்த கட்டளையை மேற் கொண்டு செய்வதுதான். மேலும் கூறப்போனால் நமது சிறுவன் கட்டளைக் கொத்தை ஏற்றபின் அவனது செயற்பாட்டில் நாம் தலையிட வேண்டியதில்லை. கட்டளைக் கொத்தை அளித்தவுடன் நமது வேலை முடிந்து விடுகிறது. இதே முறையில் தான் கம்ப்யூட்டரும் இயங்குகிறது. எந்த ஒரு கணிப்பிற்கும் அதற்குரிய கட்டளைக் கொத்தைத் தயாரித்துக் கம்ப்யூட்டருக்கு அளித்தபின் அக்கட்டளைக் கொத்தின் கட்டளைகள் யாவும் தாமாகவே ஒன்றன்பின் ஒன்றாக நிறைவேற்றப்படுகின்றன. இதையே நாம் 'தன்னியக்கத்தன்மை' என்று குறிப்பிடுகிறோம். ஒரு எடுத்துக் காட்டைப் பார்ப்போம்.

$x = 0.5$

$y = 1.0$

$z = 4x^2 + 2y - 1$ என்ற தொடர்பைப் பயன்படுத்தி zன் மதிப்பைக் கணக்கிட விரும்புவதாக வைத்துக் கொள்ளுவோம். இதைச் செய்வதற்கு ஈடான கட்டளைக் கொத்தாகக் கீழ்வரும் ஆணைகளைக் கொள்வோம்.

(i) xன் மதிப்பு 0.5 எனக் கொள்.
(ii) yன் மதிப்பு 1.0 எனக் கொள்.

(iii) xஐ xஆல் பெருக்கு
(iv) (iii)ல் கிடைத்த விடையை 4ஆல் பெருக்கு.
(v) yஐ இரண்டால் பெருக்கு.
(vi) (iv)ன் விடையையும் (v)ன் விடையையும் கூட்டு.
(vii) (vi)ன் விடையிலிருந்து 1ஐக் கழி.
(viii) (vii)ன் விடையை zன் மதிப்பெனக் கொள்.
(ix) zன் மதிப்பை வெளியிடு.

இந்த ஒன்பது கட்டளைகளையும் கம்ப்யூட்டர் புரிந்து கொள்ளும் வகையில் நாம் தயாரித்து அளிப்பின், கம்ப்யூட்டர் இவற்றை ஏற்று நினைவில் இருத்திக் கொண்டு செயல்பட ஆரம்பிக்கிறது. முதல் கட்டளையில் ஆரம்பித்து, தானாகவே ஒன்றன்பின் ஒன்றாக நினைவிலிருந்து கொணர்ந்து எல்லா கட்டளைகளையும் செய்து முடிக்கிறது. கடைசி கட்டளையை செய்து முடித்ததும் நமக்குத் தேவையான விடை, அதாவது zன் மதிப்பு நமக்கு கிடைக்கிறது. இந்த எடுத்துக் காட்டிலிருந்து நாம் வேறு சிலவும் அறியக் கூடும். அவையாவன:

(i) நமது கணக்கைச் செய்வதற்கான வழிமுறைகள் ஐயமற நமக்குத் தெரிந்திருக்க வேண்டும். ஏனெனில் நமக்கு வழிமுறை தெரியாவிடின், அதைச் செய்வதற்கான கட்டளைக் கொத்தை நாம் தயாரிக்க இயலாது.

(ii) நாம் கணக்கைச் செய்யும் வழி முறைகளை மட்டுமே கூறுகிறோம். உண்மையான கணிப்பை (Computation) கம்ப்யூட்டர் செய்கிறது.

(iii) நாம் கொடுத்த வழிமுறைகளைக் கொண்டுதான் கம்ப்யூட்டர் கணிப்பைச் செய்கிறது. நமது வழிமுறைகளிலோ, கட்டளைக் கொத்திலோ தவறு இருப்பின் கணிப்பின் முடிவு தவறாகத்தான் இருக்கும். இந்தத் தவறுக்குப் பொறுப்பு, கணிப்பிற்கான வழிமுறை வகுத்தவரோ அல்லது கட்டளைக் கொத்து தயாரித்தவரோ அன்றிக் கம்ப்யூட்டர் அல்ல.

எடுத்துக்காட்டாக, புத்தகம் வாங்கிவர நமது சிறுவனுக்காக நாம் முன் தயாரித்த ஆறு கட்டளைகளைக் கொண்ட கட்டளைக் கொத்தைப் பார்ப்போம். மேலெழுந்தவாரியாகப் பார்க்கப் போனால் அக்கட்டளைக் கொத்து முற்றும் சரியானதாகத்தான் தோன்றும். ஆனால் அது உண்மையல்ல. நமது சிறுவன் ஒரு கம்ப்யூட்டரைப் போல் இயங்குவதாகக் கொள்ளுவோம்,

அதாவது அவனுக்கு நாம் கொடுத்துள்ள கட்டளைகளை ஒன்றன் பின் ஒன்ருக நிறைவேற்றமட்டுமே தெரியும். அவன் தானுக எந்த புதிய (நாம் கூருத) செயலையும் செய்ய மாட்டான்; தானுக எந்த முடிவையும் (decision) மேற்கொள்ள மாட்டான்.

இதைக் கருத்தில் கொண்டு பையனுக்கு அளிக்கப்பட்ட கட்டளைக் கொத்தைக் காண்போமாயின் அதில் உள்ள தவறுகள் நமக்குத் தெரிய வரும். எடுத்துக்காட்டாக இரண்டாவது, மூன்ருவது கட்டளைகளை எடுத்துக் கொள்ளுவோம். இரண்டாம் கட்டளையை நிறைவேற்றிய **உடனேயே** அவன் மூன்ரும் கட்டளையை நிறைவேற்ற ஆரம்பிக்க வேண்டும். அதாவது இடது புறமாக 200 மீட்டர் தொலைவு நடந்த **உடன்** தெருவைக் குறுக்காகக் கடந்து செல்ல வேண்டும். இதன்படி அவன் தெருவைக் குறுக்காக கடக்க ஆரம்பிக்கும் போது வண்டி ஏதும் வந்தாலும் அதைப் பொருட்படுத்தாமல் அவன் தெருவைக் கடந்தாக வேண்டும். ஏனெனில் வண்டி போக்கு வரத்தைக் கவனித்து தெருவைக் கடக்குமாறு அவனுக்கு கட்டளை இடப் படவில்லை. இந்த விதத்தில் பார்க்கப் போனுல், நமது சிறுவனுக்கு நாம் தயாரித்துள்ள கட்டளைக் கொத்து 'மிகவும் ஆபத்தானதும் கூட.'

ஒரு கணிப்பிற்கான கட்டளைக் கொத்தைத் தயாரிக்கும் போது அந்தக் கணிப்பினூடே ஏற்படக்கூடிய எல்லா விதமான நிகழ்வுகளையும் கருத்தில் கொண்டு, அத்தகைய நிகழ்வுகளில் ஒவ்வொன்றும் ஏற்படும் போது என்னென்ன செய்ய வேண்டும் என்பதையும் விரிவாகவும் தெளிவாகவும் ஐயத்திற்கிடமின்றி யும் கட்டளைக் கொத்து கூற வேண்டும். அப்போதுதான் அது ஒரு முழுமை பெற்ற (wholesome) கட்டளைக் கொத்து ஆகும்.

கம்ப்யூட்டர் மொழி

கடைக்குச் சென்று புத்தகம் வாங்கி வரத் தேவையான கட்டளைகளை நமது சிறுவனுக்கு நாம் இடவேண்டுமாயின், அதற்கு ஒரு மொழி தேவைப்படுகிறது; அந்த மொழி சிறு வனுக்குத் தெரிந்திருத்தல் அவசியம்—இதையே வேறு விதமாகக் கூறினுல் நாம் இடும் கட்டளைகள் சிறுவனுக்குத் தெரிந்த மொழி யில் இருத்தல் வேண்டும். இதே போல் கம்ப்யூட்டரும் நமது ஆணைகளைப் 'புரிந்து' கொள்ள வேண்டுமாயின் முதலாவதாக கம்ப்யூட்டருக்கு ஒரு மொழி தெரிந்திருக்க வேண்டும். அடுத்து நமது கட்டளைகள் யாவும் கம்ப்யூட்டர் 'அறிந்த, மொழியில் இருக்க வேண்டும். இன்றைய கம்ப்யூட்டரின் வடிவமைப்

பிலேயே அதற்கென்று ஒரு மொழி உருவாகிறது. இதைக் கம்ப்யூட்டரின் 'பிறவி மொழி' என்று கூறலாம். எவ்வாறு மனித மொழிகள் இனத்திற்கு இனம் அல்லது இடத்திற்கு இடம் வேறுபடுகின்றனவோ, அவ்வாறே கம்ப்யூட்டர்களின் மொழிகளிலும் ஒன்றுக்கொன்று வேறுபாடு உண்டு; எடுத்துக்காட்டாக IBM *1620* கம்ப்யூட்டரின் மொழிக்கும் IBM *360* கம்ப்யூட்டரின் மொழிக்கும் வேறுபாடு உண்டு; ICL *1900* கம்ப்யூட்டரின் மொழி இவையிரண்டிலிருந்தும் மாறுபட்டிருக்கும்.

மனித மொழிகளுக்கும், கம்ப்யூட்டர் மொழிகளுக்கும் பலவித வேறுபாடுகள் உண்டு. மனித மொழிகளைவிட, கம்ப்யூட்டர் மொழி மிகவும் 'சிக்கனமானது.' எடுத்துக் காட்டாக அடைமொழி, காலம், ஒருமை, பன்மை, பால் பாகுபாடு போன்றவை கம்ப்யூட்டர் மொழிகளில் கிடையா. ஒவ்வொரு வாக்கியத்திலும் வினைச் சொல் ஒன்றும் அதற்கு இன்றியமையாத செயற்படு—பொருள் மட்டுமே காணலாம். கம்ப்யூட்டரின் செயல் வகைகள் எண்ணிக்கையில் மிகவும் குறைந்தனவாயிருப்பதால் (மனிதர்களின் செயலாற்றல்களின் எண்ணிக்கையை ஒப்ப) தனித்தனி வாக்கியங்களின் எண்ணிக்கையும் அதற்கொப்ப குறைந்தே காணப்படும். வாக்கியங்களின் விகுதிகளை மட்டும் மாற்றியும், பயன்படுத்தப்பட்ட வாக்கியங்களையே திரும்பத் திரும்ப பயன்படுத்தியுமே நமது கட்டளைக் கொத்து தயாரிக்கப் படுகிறது.

அடுத்தபடியாக, மனித மொழிகளில் ஒரு சொல்லுக்குப் பல பொருள்கள் அமைந்திருப்பதும், பல சொற்களுக்கு ஒரே பொருள் இருப்பதும் சாதாரணமாகக் காணக் கூடிய ஒன்றாகும். மனித மொழிகளைப் பொறுத்தவரை சிலேடைகள் அழகையும், புலவருக்கு ஒரு வித சௌகரியத்தையும் கொடுக்கலாம். இந்த அமைப்பு கம்ப்யூட்டரின் மொழியில் கிடையாது. அங்கே ஒரு சொல்லுக்கு ஒரே பொருளும், ஒரு பொருளுக்கு ஒரே சொல்லும் தான் உண்டு.

கம்ப்யூட்டர் துறையில் நடக்கும் பலவித ஆய்வுகளில் மிகவும் சிக்கனமான, ஆனால் அதே சமயத்தில் நடைமுறையில் கையாளக் கூடிய, கம்ப்யூட்டர் மொழிகளை அமைப்பதும் ஒன்றாகும். இம் மொழியைக் கையாள்வது பற்றியும், கையாண்டு கட்டளைக் கொத்துக்களைத் தயாரிப்பது பற்றியும் பிறிதோரிடத்தில் மீண்டும் காண்போம்.

நினைவகம்

கம்ப்யூட்டர் தனக்கிடப்படும் கட்டளைகளை முதலில் நினைவில் ஏற்றுக் கொண்டு பின்னர் அவற்றை நிறைவேற்றுகிறது என

சற்று முன்னர் பார்த்தோம். மனிதர்கள் தமக்கிடப்படும் கட்டளை களை நினைவில் கொள்ள அவர்களுக்கு மூளை இருக்கிறது. மனிதர் களைப் போல இவ்வியந்திரங்களும், கட்டளைகளை நினைவில் கொள்ள வேண்டுமாயின், அதற்கேற்ப மூளையைப் போன்று ஒரு அமைப்பு அவற்றிற்கு இருத்தல் அவசியம். இன்றைய கம்ப் யூட்டர்களுக்கு இத்தகைய 'மூளையும்' இருக்கிறது. கம்ப்யூட் டரின் இந்த பாகத்தை 'மூளை' என்று கூறுமல் 'நினைவகம்' என்று கூறுகிருர்கள். ஏனெனில் நினைவில் கொள்ளும் ஆற்றல் நீங்கலாக மூளையின் மற்ற பல்வேறு ஆற்றல்களும் இந்த நினை வகங்களுக்கு கிடையா. கம்ப்யூட்டரின் நினைவாற்றலுக்கும், மனிதனின் நினைவாற்றலுக்கும் ஓர் அடிப்படை வேறுபாடு உண்டு. மனிதமூளை, தான் கற்றவற்றை எளிதில் மறந்து விடும் இயல்புடையது. ஆனுல் கம்ப்யூட்டரின் நினைவில் செய்தியை ஒருமுறை ஏற்றினுல், கம்ப்யூட்டர் அதை என்றென்றும் நினைவில் வைத்துக் கொள்ளும். கம்ப்யூட்டர் தானுகவே எதையும் மறப் பதில்லை. அதன் நினைவில் உள்ள செய்தி முழுவதையுமோ, அல்லது ஒரு பாகத்தையோ 'மறந்து' விடுமாறு நாம் அதற்கு ஆணையிட்டால் ஒழிய அதன் நினைவில் உள்ள செய்தி சற்றும் மாருமல் என்றென்றும் இருக்கவல்லது. சாதாரணமாக, ஒரு கட்டளைக் கொத்தை ஏற்று செயல்பட்டு முடிந்த உடன், கம்ப் யூட்டருக்கு அதன் நினைவகத்தைச் சுத்திகரிக்குமாறு (clear memory), ஆணையிட்ட பிறகே மற்ெருரு கணிப்பிற்கான கட்டளைக் கொத்து அதன் நினைவில் ஏற்றப்படுகிறது. மனிதனின் மூளையில் நுழைக்கப்பட்ட செய்தியை சாதாரணமாக எந்த விதத்திலும் முழுமையாக மறக்கச் செய்ய இயலாது. ஆனுல் கம்ப்யூட்டரின் நினைவகத்திலிருந்த செய்தியை நினைத்தபோது முழுமையாக மறக்கடிக்கமுடியும். மனிதன் மூளையில் வைத்துக் கொள்ளக் கூடிய செய்திகளுக்கு அளவு கிடையாது. ஆனுல் கம்ப்யூட்டரின் மூளையில் ஒரு குறிப்பிட்ட அளவு செய்திகளைத் தான் நுழைக்க முடியும். இனி அடுத்து வரும் சில பக்கங்களில் கம்ப்யூட்டரின் நினைவகத்தின் அடிப்படை அமைப்பு பற்றியும், அது எவ்வாறு செயல் படுகிறது என்றும் பார்ப்போம்—அதற்கு முன்னர், நாம் வேறு சிலவற்றைத் தெரிந்து கொள்ள வேண்டியுள்ளது.

இருமையமைப்பு (Binary system)—**பைனரி அமைப்பு**

அன்ருட வாழ்க்கையில் நாம் காணும் சில அமைப்புக்கள் இரண்டேயிரண்டு நிலைகளில் மட்டுமே இயங்கவல்லன. எடுத்துக் காட்டாக, ஒரு விளக்கை எடுத்துக் கொண்டால், ஒன்று அது எரிந்து கொண்டிருக்கும், அல்லது அணைந்த நிலையில் இருக்கும்; ஒரு மின்காந்தத்துண்டு காந்த நிலையில் இருக்கும்; அல்லது

காந்தமற்ற நிலையில் இருக்கும்; சுவற்றிலிருக்கும் கடிகாரம் ஒன்று ஓடிக் கொண்டிருக்கும், அல்லது நின்று போயிருக்கும்; நமக்குக் கிடைத்த ஒரு செய்தி உண்மையாயிருக்கலாம் அல்லது பொய்யாயிருக்கலாம். இப்படி எத்தனையோ, இவ்வாறு இரண்டு நிலைகளில் மட்டும் இயங்கும் அமைப்புக்களையே நாம் இருமை யமைப்பு (Binary system) என்று கூறுகிறோம்.

கம்ப்யூட்டர்களின் இயக்கம் முழுதும் மின்சாரத்தால் தான் நடைபெறுகிறது. கம்ப்யூட்டரின் உள்ளமைப்புகளில் உள்ள பாகங்களின் வழியாக மின்சாரத்தைச் செலுத்தியோ அல்லது நிறுத்தியோ தான் பலவேறு செயல்களையும் செய்ய வேண்டியுள்ளது. அதாவது கம்ப்யூட்டரின் இயக்கம் முழுதும் இருமை அமைப்பின் அடிப்படையிலேயே நடக்க வேண்டியுள்ளது. கட்டளைக் கொத்தின் கட்டளைகளாகட்டும், கணிப்பிற்கு வேண்டிய அடிப்படை விவரங்களாகட்டும் (Data) யாவுமே இருமை முறையைக் கொண்டே கம்ப்யூட்டரின் நினைவில் இருத்தியாக வேண்டும். இது எவ்வாறு என்று சுருக்கமாகப் பார்ப்போம்.

இருமைக்குறி தசம எண் (Binary Coded Decimal Digit)

கீழுள்ள படம் 1.2ல் B_0, B_1, B_2, B_3, என்பன ஒரே மாதிரி யான நான்கு மின்விளக்குகள். இம்மின் விளக்குகள் ஒவ் வொன்றையும் எரி நிலை (ON) யிலேயோ அல்லது அணை நிலை (OFF)யிலேயோ வைக்கலாம். இம்மாதிரி இருமை நிலையில் இருக்க வல்ல ஒவ்வொரு அமைப்பையும் ஆங்கிலத்தில் 'பிட்' (Bit) என்று கூறுவார்கள். (இது Binary Digit என்ற சொல்லின் சுருக்க மாகும்). இங்குள்ள ஒவ்வொரு பல்பும் ஒரு 'பிட்' ஆகும்.

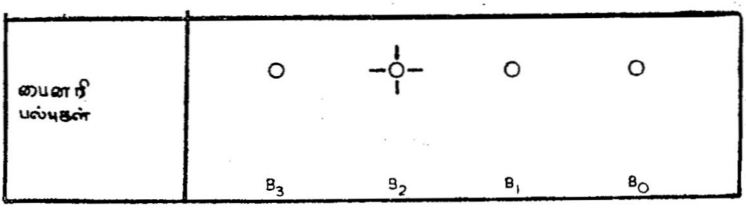
எரிநிலை மதிப்பு	8	4	2	1
பல்புகள்	⊙ B3	⊙ B2	⊙ B1	⊙ B_0
அணை - நிலை மதிப்பு	0	0	0	0

படம் 1.2

தசம இலக்கத்திற்கான பைனரி அமைப்பு
(A binary set-up for a decimal digit)

இந்த விளக்குகளில் Bo எரிந்தால் அதற்கு 2^0 மதிப்பும் (அதாவது 1) அணைந்த நிலையில் 0 மதிப்பும் கொடுப்போம்.

அதே போல் B_1 எரிந்தால் 2^1 (அதாவது 2 மதிப்பும்) அணைந்திருந்தால் 0 மதிப்பும் கொள்ளுவோம். அவ்வாறே B_2, B_3, இவைகளும் எரியும் போது முறையே 2^2 (அதாவது 4), 2^3 (அதாவது 8) மதிப்பையும், அணைந்த நிலையில் ஒவ்வொன்றுக்கும் 0 மதிப்பையும் அளிப்போம். இந்த மதிப்புகளைக் கொண்டு, படத்திலுள்ள நான்கு விளக்குகளில் தேவையானவற்றை ஏற்றியும், மற்றவற்றை நிறுத்தியும் 0 முதல் பதினைந்து வரை உள்ள எல்லா எண்களையும் குறிப்பிடலாம். எடுத்துக்காட்டாக 4 என்ற எண்ணைக் குறிக்க B_2-வை மட்டும் எரிய வைத்து மற்றவற்றை அணைத்து விடலாம்.



படம் 1.3

4—ஐ குறிக்கும் பைனரி அமைப்பு

7 என்ற எண்ணைக் குறிக்க, B_0, B_1, B_2, என்ற மூன்றையும் ஏற்றி B_3ஐ அணைத்து வைக்கலாம். B_0 எரிவதால் அதன் மதிப்பு 1 ஆகவும், B_1 எரிவதால் அதன் மதிப்பு 2 ஆகவும், B_2 எரிவதால் அதன் மதிப்பு 4 ஆகவும், B_3 அணைந்திருப்பதால் அதன் மதிப்பு 0 ஆகவும் ஆகிறது. இவையெல்லாவற்றையும் கூட்ட $0+4+2+1=7$ ஆகிறது.

இவ்வமைப்பில் எல்லா விளக்குகளையும் அணைத்து வைத்தோமானால் மொத்த மதிப்பு 0 ஆகிறது. இவ்வாறே பலவேறு விளக்குகளை ஏற்றியும், நிறுத்தியும் 0 முதல் 9 வரை உள்ள எல்லா எண்களையும் நாம் குறிப்பிட இயலும். படம் 1.4ஐக் காண்க.

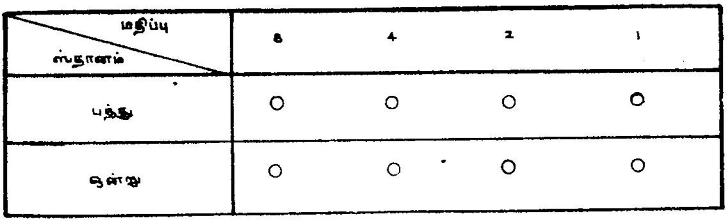
படம் 1—4ல் எவ்வாறு 0 முதல் 9 வரை உள்ள எண்களை (அதாவது ஒற்றை இட எண்களை) நான்கு விளக்குகளைக் கொண்ட ஒரு அமைப்பைக் கொண்டு குறிக்கலாம் என்று காட்டப்பட்டுள்ளது. ஒன்றைப்படை எண்ணைக் குறிக்க, நாம் நான்கு விளக்குகளைக் கொண்ட ஒரு தொகுதியைப் பயன்படுத்தினோம். அதற்கு மாறாக, இப்போது ஒவ்வொன்றிலும் நான்கு பல்புகளை உடைய இரண்டு தொகுதிகளை எடுத்துக் கொண்டு, ஒரு தொகுதி

யைப் பத்தாவது இடத்திற்கும், மற்றதை ஒன்றுவது இடத்திற்கும் ஒதுக்குவோம். படம் 1.5ல் இந்த அமைப்பு காட்டப் பட்டுள்ளது.

தசம இலக்கம்	விளக்குகளின் நிலை			
	8	4	2	1
0	○	○	○	○
1	○	○	○	☼
2	○	○	☼	○
3	○	○	☼	☼
4	○	☼	○	○
5	○	☼	○	☼
6	○	☼	☼	○
7	○	☼	☼	☼
8	☼	○	○	○
9	☼	○	○	☼

படம் *1.4*

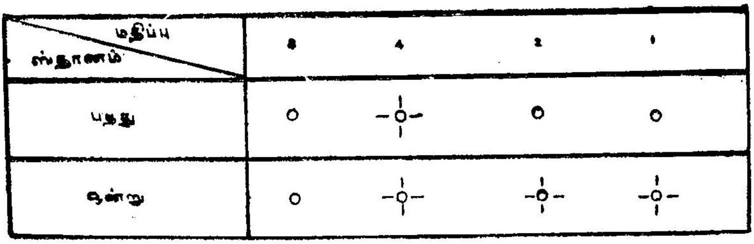
தசம இலக்கங்களும் அவற்றிற்கான பைனரி அமைப்பும்



படம் 1.5

ஈரிலக்க தசம எண்களைக் குறிக்கும் பைனரி அமைப்பு

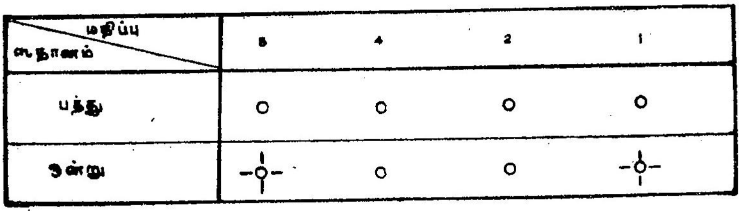
மேற்காணும் இந்த அமைப்பைக் கொண்டு இரண்டு இட எண்களை எவ்வாறு குறிப்பது ? எடுத்துக்காட்டாக 47 என்ற எண்ணை எடுத்துக் கொள்ளுவோம். இங்கு 4 என்ற எண் பத்தாவது இடத்திலும், 7 என்ற எண் ஒன்ருவது இடத்திலும் உள்ளன. அதனுல் பத்துக்கு உரிய விளக்குத் தொகுதியைக் கொண்டு 4-யும் ஒன்றுக்குரிய விளக்குத் தொகுதியைக் கொண்டு 7-யும் கீழ் கண்டவாறு குறிக்கலாம்.



படம் 1.6

'47'-ஐக் குறிக்கும் பைனரி அமைப்பு

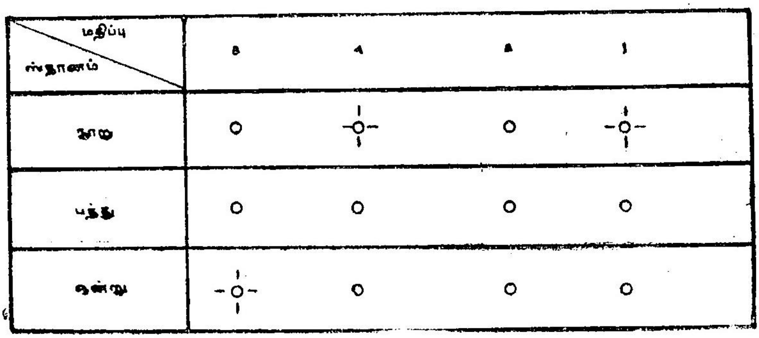
படம் 1—6-ல் 47—என்ற எண்ணை எவ்வாறு நான்கு விளக்குகளை உடைய இரண்டு தொகுதிகளைக் கொண்டு குறிப்பிட இயலும் என்று காட்டப்பட்டுள்ளது. இதைப் போலவே மற்ற எல்லா இரண்டு இட எண்களையும் பத்து, ஒன்று ஆகிய தொகுதிகளில் தகுந்த விளக்குகளை ஏற்றியும், நிறுத்தியும் குறிப்பிடலாம். படம் 1-6ல் உள்ள அமைப்பைக் கொண்டு ஒன்ருவது இட எண்களையும் நாம் குறிக்க இயலும். ஒன்ருவது இட எண்களை குறிக்கும் போது பத்தாவது இட விளக்குகள் நான்கையும் நிறுத்திக் கொண்டு, ஒன்ருவது இட விளக்குகளில் தகுந்தவற்றை ஏற்றிக் கொள்ள வேண்டும். எடுத்துக்காட்டாக 9 என்ற ஒற்றை இட எண்ணை கீழுள்ளவாறு குறிக்கலாம். மேற் கூறிய



படம் 1.7

இரட்டை யிலக்க தசம எண்களைக் குறிக்கும் பைனரி அமைப்பு '9' என்ற ஓரிலக்க எண்ணைக் குறிக்கிறது

முறையைப் பின் பற்றி நாம் மற்றும் ஓர் நான்கு விளக்கு தொகுதியை நூறுவது இடத்திற்காக சேர்த்துக் கொண்டால், 000 முதல் 999-வரை உள்ள எண்களை எளிதில் குறிக்க இயலும். 508 என்ற எண்ணை குறிக்கும் கீழுள்ள அமைப்பைக் காண்க.



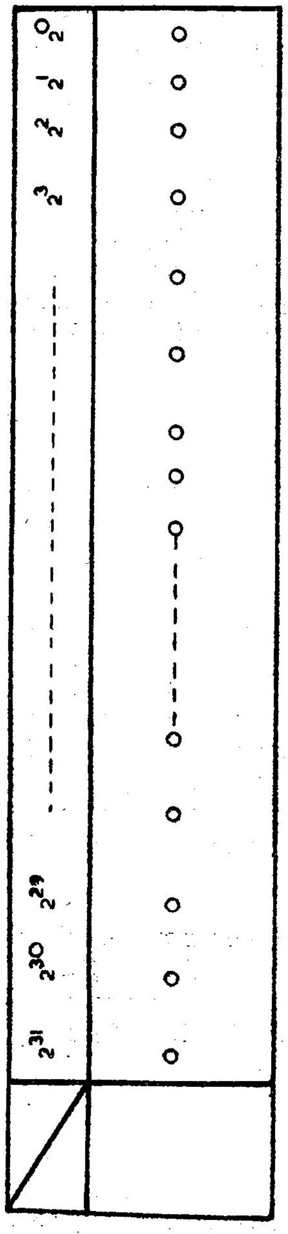
படம் 1.8

'508'-ஐக் குறிக்கும் பைனரி அமைப்பு

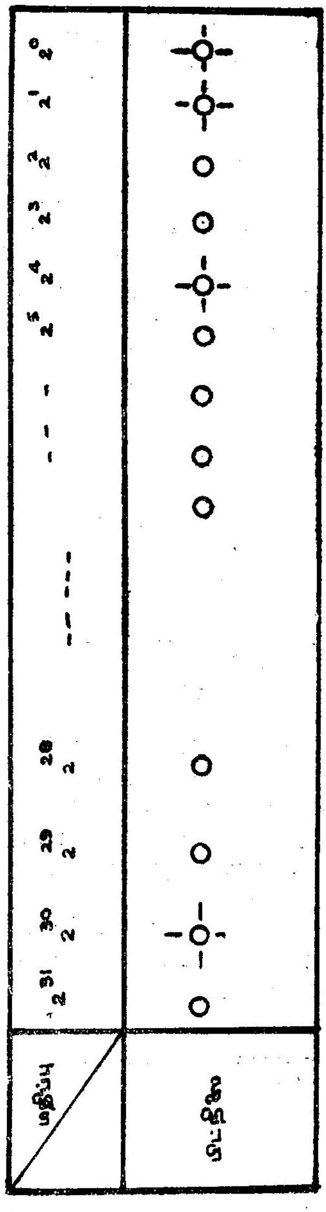
இவ்வாறே விளக்குத் தொகுதிகளின் எண்ணிக்கையை வேண்டிய அளவு அதிகரித்து மேற்கூறிய முறையில் 4, 5, 6...... இட எண்களையும் நாம் எளிதில் குறிக்க இயலும்.

இந்தப் பகுதியில் இதுவரை நாம் செய்தது என்ன ?

மின்விளக்கு ஒவ்வொன்றும், ஒரு இருமை அமைப்பு (Binary) ஆகும். நான்கு விளக்குகளைக் கொண்டு— அதாவது நான்கு இருமை அமைப்புகளைக் கொண்டு— ஒரு தசம இலக்கத்தை (decimal digit) நாம் குறிப்பிட்டிருக்கிறோம். இரண்டு இட எண்களைக் குறிப்பிடும் அமைப்புகளிலும் (படம் 1-5 முதல் படம் 1-7



படம் 1.9
பைனரி முறை எண் குறிப்பு



படம் 1.10
1141899283 ($= 2^{30} + 2^{4} + 2^{1} + 2^{0}$) என்ற எண்ணைக் குறிக்கும் பைனரி அமைப்பு

முடிய) ஒற்றை இடத் தொகுதி விளக்குகளுக்கும், பத்து ஸ்தான தொகுதி விளக்குகளுக்கும், அமைப்பில் வேறுபாடு இல்லை. பத்தாவது இடத் தொகுதிக்கு, இட மதிப்பு கொடுத்துள்ளோம்; அவ்வளவே. எடுத்துக்காட்டாக 79 என்ற இரண்டு இட எண்ணில் 7, 9, என்ற இரண்டு எண்கள் (digits) உள்ளன. இவற்றுள் 7 என்ற எண்ணுக்கு ஸ்தான மதிப்பு 10 ஆக இருப்பதால் அது ஏழு பத்து ஆகி ஒன்பதுடன் சேர்ந்து எழுபத்து ஒன்பதாகிறது. இதே முறையில் தான் படம் (1-6)ல் பத்து வரிசையில் உள்ள 2^2 விளக்கு 2^2—அதாவது நான்கு—பத்துகளை குறிக்கிறது. இந்த நான்கு பத்துகளுடன் கீழ் வரிசை (ஒன்று வரிசை)யில் உள்ள ஏழும் சேர்ந்து நாற்பத்தேழாகிறது. ஆக நாம் செய்திருப்பது, 'பிட்'களைக் கொண்டு தசம எண்களைக் குறித்திருக்கிறோம். இவ்வாறு குறிக்கப்படும் எண்களை 'இருமைக்குறி தசம எண்' (Binary coded decimal digit) என்று கூறுவார்கள்.

மேற்கூறிய முறையில் 'இரு நிலை அமைப்பு' தசம இட மதிப்பு ஆகிய இரண்டு அடிப்படைகளைக் கொண்டு எண்களை நாம் குறித்தோம். இம்முறையிலிருந்து மாறுபட்ட அமைப்புக் களும் உண்டு. எடுத்துக் காட்டாக 'இருமைக்குறி எண்' முறையைக் கூறலாம். இம்முறையில் 'தசம இட மதிப்பு' கிடையாது; இருமை இட அமைப்பு மட்டுமே உள்ளது. முன் குறித்த முறையில் நமது குறியீட்டில் (coding) நான்கு விளக்குகளைக் கொண்ட தொகுதிகள் அடிப்படையாக அமைந்திருந்தன. இதற்கு மாருக 32 விளக்குகளை (பிட்களைக்) கொண்ட ஒரு அமைப்பை எடுத்துக் கொண்டு, பல்புகளுக்கு எரி நிலையில் முறையே (வலமிருந்து இடமாக) 2^0, 2^1, 2^2, 2^3,...2^{30}, 2^{31} என்ற மதிப்புகளைக் கொடுப்போம். அணை நிலையில் முன் போல் ஒவ்வொரு பல்புக்கும் o மதிப்பு கொடுப்போம். படம் 1-9ல் இத்தகைய அமைப்புக் காட்டப் பட்டுள்ளது.

படம் 1-9ல் உள்ள 32 'பிட்'களை தகுந்தவற்றை ஏற்றியும், அணைத்தும் பல் வேறு எண்களையும் நாம் குறிப்பிட இயலும். எடுத்துக் காட்டாக 2^{30}, 2^4, 2^1, 2^0, ஆகிய பல்புகள் எரிந்தும், மற்றவை அணைந்தும் இருப்பின் இது 1141899283 (அதாவது $2^{30}+2^4+2^1+2^0 = 1141899264+16+2+1 = 1141899283$) என்ற எண்ணைக் குறிக்கும்.

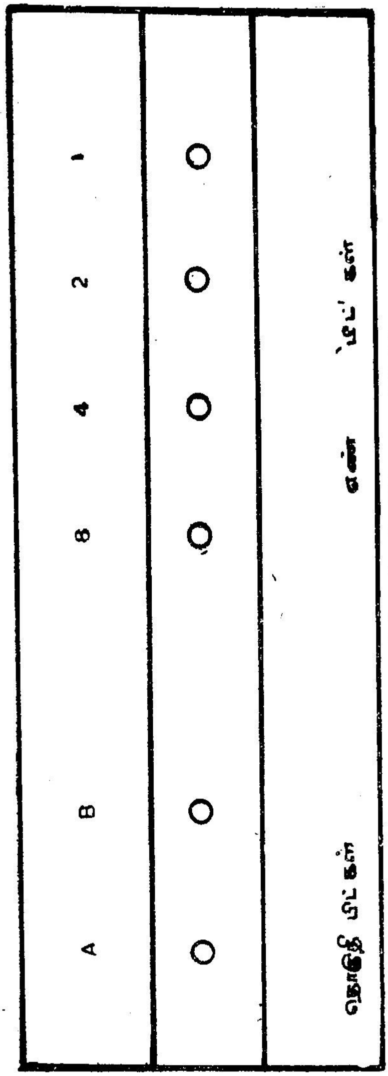
இந்த முறைக்கு இரு நிலை எண்குறிமுறை (Binary coded digit) என்று கூறுவர். இந்த முறைக்கும் முன் குறித்த இரு நிலைக் குறி தசம எண் குறியீட்டு முறைக்கும் உள்ள முக்கிய

வேறுபாடுகளில் குறிப்பிடத்தக்கது 'பிட்'களின் எண்ணிக்கை. எடுத்துக்காட்டாக படம் (1-10)-ல் காட்டப்பட்டுள்ள 1141899283 என்ற எண்ணை எடுத்துக் கொள்ளுவோம். இதை குறிக்க (இரு நிலை எண் குறி முறையில்) 32 'பிட்'களைக் கொண்ட ஒரே ஒரு தொகுதி இருக்கிறது. இதே எண்ணை இருமைக்குறி தசம எண் முறையில் குறிக்க வேண்டுமானால் நமக்கு $10 \times 4 = 40$ 'பிட்'கள் தேவைப்படும். இன்று கையாளப்படும் கம்ப்யூட்டர் களில் சில வகைகளில் முதல் முறையும் (இருமைக் குறி தசம எண் முறையும்) மற்றும் சிலவற்றில் இருமை முறையும் பயன் படுத்தப்படுகின்றன. இன்று வடிவமைக்கப்படும் பெரும்பாலான கம்ப்யூட்டர்கள் இரண்டாவது முறையில் அமைந்த குறி யீட்டையே பயன்படுத்துகின்றன. இருப்பினும் நம் கற்கும் முறையின் எளிமை குறித்து நாம் முதல் முறையையே—அதா வது, இருமைக் குறி எண் முறையையே அடுத்து வரும் பக்கங் களில் பின்பற்றுவோம்.

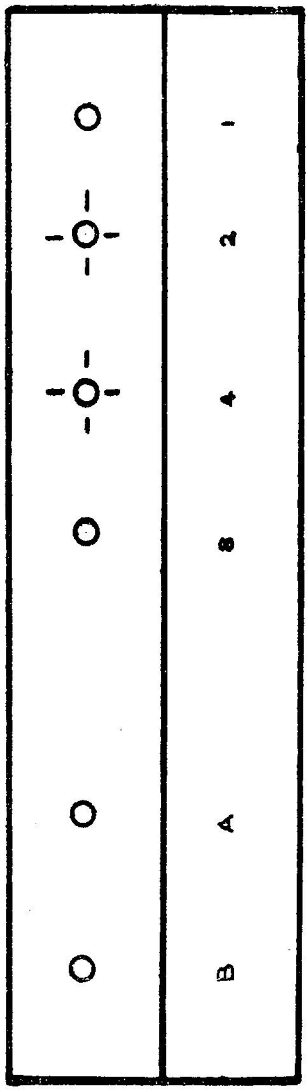
இதுவரை நாம் பார்த்தவை யாவும் பைனரி என்ற இரு நிலை முறையை (அல்லது இருமைக்குறி தசம எண் முறையை) பயன்படுத்தி எண்களை குறிப்பிடும் வழியாகும். ஆனால் இன் றைய கம்ப்யூட்டர்கள் எண்களை மட்டும் கையாள்வது இல்லை; A, B, C, D...போன்ற எழுத்துக்களையும் *, $, +, — போன்ற குறிகளையும் பயன்படுத்துகின்றன.

அதனால் எண்கள் மட்டுமின்றி, இவை போன்ற குறியீடு களையும் இருமை முறையில் குறிப்பிடவேண்டிய தேவை ஏற்படு கிறது. நடைமுறயில் இதற்காக ஒன்றுக்கு மேற்பட்ட முறைகள் கையாளப்படுகின்றன. இவற்றுள் ஒரே ஒரு முறையைக் காண் போம். எண்களுடன், எழுத்துக்களையும், மற்றக் குறிகளையும் கையாளுதல் வேண்டி முன் குறித்த நான்கு 'பிட்'களுடன் இன்னும் இரண்டு 'பிட்'களையும் நாம் இணைப்போம். இவை யிரண்டும் 'பகுதி-பிட்'கள் (zone bits) எனவும், 2^0 முதல் 2^3 வரை உள்ள நான்கு 'பிட்'களைக் கொண்ட தொகுதிக்கு 'எண்-பிட்' (Numeric bits) தொகுதி என்றும் பெயரிடுவோம்.

படம் 1.11-ல் பகுதி 'பிட்'கள் இரண்டுக்கும் முறையே A, B என்ற பெயர்கள் கொடுக்கப் பட்டுள்ளன. இவ்வமைப்பைக் கொண்டு எண்களை மட்டுமல்லாமல் மற்ற எழுத்துக்கள், குறிகள் ஆகிய வற்றையும் நாம் குறிக்கலாம்.



படம் 1.11 கம்ப்யூட்டர் நினைவக 'பிட்' அமைப்பு



படம் 1.12 6-ஐக் குறிக்கும் 'பிட்' அமைப்பு

A,B ஆகிய இரண்டு 'பிட்'களையும் அணைத்து வைத்து, எண் 'பிட்' தொகுதியில் தகுந்த 'பிட்'களை ஏற்றி *0* முதல் *9* வரை உள்ள எண்களைக் குறிப்பிடலாம்; எழுத்துக்களையும் மற்ற குறியீடுகளையும் குறிப்பிட 'பகுதி-பிட்' தொகுதி, 'எண்-பிட்' தொகுதி ஆகிய இரண்டையும் பயன் படுத்தலாம். சில எடுத்துக்காட்டுகள் படங்கள் *1.12; 1.13; 1.14* ஆகியவைகளில் காட்டப் பட்டிருக்கின்றன.

மேற்கண்ட எடுத்துக் காட்டுகள் உண்மையாகவே ஒரு கம்ப்யூட்டரில் (IBM *1401*) பயன் படுத்தும் 'பிட்' குறியீடுகள் ஆகும். இக்கம்ப்யூட்டர் பயன் படுத்தும் எண்களும், எழுத்துக்களும் பல்வேறு குறிகளும் அவற்றிற்கான இருமைக்குறியமைப்புகளும் அட்டவணை 1-2ல் கொடுக்கப் பட்டுள்ளன. இந்த அட்டவணையில் *1*-என்ற எண் எரியும் 'பிட்'களையும், *0* என்ற எண் அணைந்த நிலையில் உள்ள 'பிட்'களையும் குறிப்பிடுகின்றன.

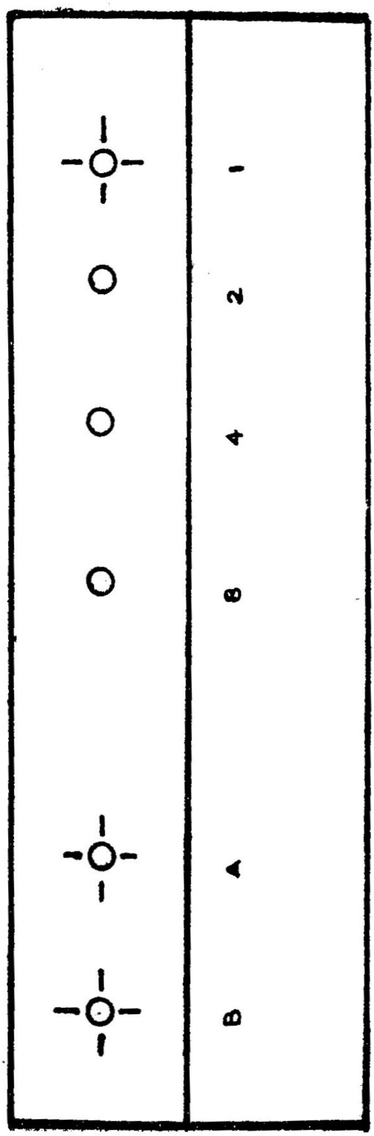
கணிப்புகளின் ஊடே, கம்ப்யூட்டரின் நினைவகத்தின் ஓரிடத்தில் உள்ள எண்கள், பிரிதோரிடத்திற்கு அடிக்கடி கொண்டு செல்லப் படுகின்றன. அதுவுமல்லாமல் நினைவகத்திலிருந்து மற்ற பகுதிகளுக்கும் இவ்வெண்களும், குறியீடுகளும் அடிக்கடி இடம் பெயர்ந்து செல்ல வேண்டியிருக்கும். இவ்வாறு இடம் பெயர்ந்து செல்லும் போது, சில சமயங்களில் தவறுகள் நேரக் கூடும். ஏதோ ஒரு சிறு மின்கம்பி (wire) தளர்வுற்று இருந்தாலோ, அல்லது வேறு பிற காரணங்களாலோ, ஒரு குறியீடு நினைவகத்தில் ஓரிடத்திலிருந்து மற்றோரிடத்திற்குச் செல்லும் போது வேறொரு குறியீடாகத் தவறி விட ஏதுவாகும். எடுத்துக் காட்டாக, படம் *1-15*-ல் உள்ள குறியீடு, நினைவகத்தின் ஒரு இடத்திலிருந்து மற்றொரு இடத்திற்குக் கொண்டு செல்லப் படுவதாகவைத்துக் கொள்ளுவோம். அவ்வாறு கொண்டு செல்வதற்கு மின் துடிப்புகள் (Electric pulses) பயன்படுகின்றன. சில கோளாறுகளால் A-'பிட்'டின் நிலையைக் கொண்டு செல்ல வேண்டிய மின் துடிப்பு வலிவு குன்றி இருப்பின் அது எரிநிலைக்கு மாருக, அணைநிலையாக புதிய இடத்தில் படம் *1-16*-ல் உள்ளவாறு வடிவெடுக்கும். அதாவது முதலில் % ஆக இருந்தது புதிய இடத்தில் @ என்ற தவருன குறியாக படம் *1-16*-ல் உள்ளது போல் உருவெடுக்கிறது. (அட்டவணை 1.2ஐ நோக்குக)

இதேபோல் ஓரிடத்தில் அணை நிலையில் இருக்கும் 'பிட்'டை மற்றோரிடத்தில் பெயர்த்து வைக்கும் போது அது எரி நிலைக்கு

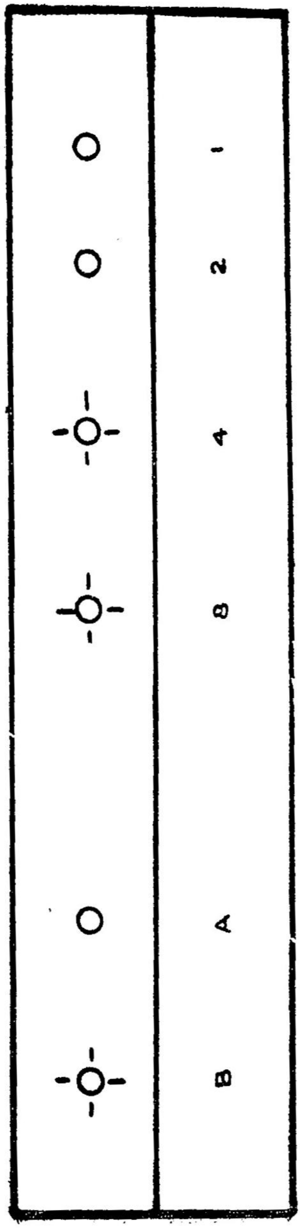
IBM *1401*-ன் குறிகளும் அவற்றின் இருமை (பைனரி)க் குறி யமைப்புகளும்

1401 குறி	பைனரி	கார்டுக் குறி	*1401* குறி	பைனரி	கார்டுக் குறி
0	1 00 1010	0	O	0 10 0110	11-6
1	0 00 0001	1	P	1 10 0111	11-7
2	0 00 0010	2	Q	1 10 1000	11-8
3	1 00 0011	3	R	1 10 1001	11-9
4	0 00 0100	4	S	1 01 0010	0-2
5	1 00 0101	5	T	0 01 0011	0-3
6	1 00 0110	6	U	1 01 0100	0-4
7	0 00 0111	7	V	0 01 0101	0-5
8	0 00 1000	8	W	0 01 0110	0-6
9	1 00 1001	9	X	1 01 0111	0-7
A	0 11 0001	12-1	Y	1 01 1000	0-8
B	0 11 0010	12-2	Z	0 01 1001	0-9
C	1 11 0011	12-3	&	1 11 0000	12
D	0 11 0100	12-4	.	0 11 1011	12-3-8
E	1 11 0101	12-5	□	1 11 1100	12-4-8
F	1 11 0110	12-6	—	0 10 0000	11
G	0 11 0111	12-7	$	1 10 1011	11-3-8
H	0 11 1000	12-8	*	0 10 1100	11-4-8
I	1 11 1001	12-9	/	1 01 0001	0-1
J	1 10 0001	11-1	,	1 01 1011	0-3-8
K	1 10 0010	11-2	%	0 01 1100	0-4-8
L	0 10 0011	11-3	#	0 00 1011	3-8
M	11 0 0100	11-4	@	1 00 1100	4-8
N	01 0 0101	11-5	blank	1 00 0000	

அட்டவணை *1.2*



படம் 1.13 A-ஐக் குறிக்கும் 'பிட்' அமைப்பு

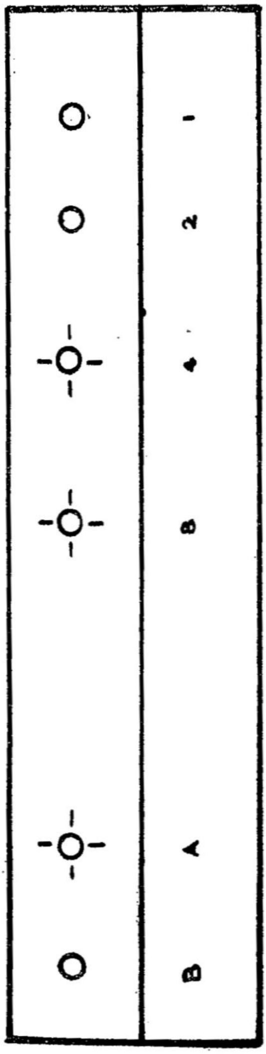


படம் 1.14 '*'-ஐக் குறிக்கும் 'பிட்' அமைப்பு

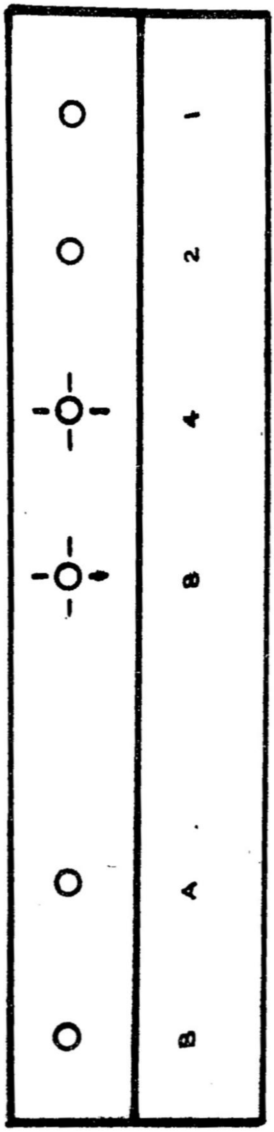
வந்து விடவும் கூடும். இத்தகைய தவறுகள் ஏற்படாமல் பாது காத்துக் கொள்ளுவது மிகவும் இன்றியமையாதது. அதன் பொருட்டு நாம் முன்னர் பயன்படுத்திய ஆறு 'பிட்'களுடன் மேலும் ஒரு 'பிட்'டையும் சேர்த்திருக்கிறார்கள். இதற்கு 'சோதனைபிட்' (Check bit or parity bit) என்று பெயரிடப்பட்டுள்ளது. இது பயன்படும் விதம் வருமாறு: 'சோதனை-பிட்'டையும் சேர்த்து இப்போது நமக்கு ஏழு 'பிட்' கொண்ட தொகுதி உள்ளது. ஏதாவது ஒரு எழுத்தை அல்லது குறியீட்டைக் குறிக்கும் போது சோதனை பிட் நீங்கலாக, எரிநிலையில் உள்ள 'பிட்'களின் எண்ணிக்கை ஒற்றைப்படை எண்ணாக இருந்தால் சோதனை பிட் அணைநிலையில் இருக்குமாறும், அதற்கு மாறாக எரிநிலையில் உள்ள 'பிட்'களின் எண்ணிக்கை (சோதனை பிட் நீங்கலாக) இரட்டைப் படை எண்ணாக இருப்பின் 'சோதனை பிட்' தானாகவே எரிநிலைக்கு வருமாறும் அமைக்கப் பட்டுள்ளது. ஆக எந்த குறியீட்டை நாம் குறிப்பதாயினும்— எண், எழுத்து இன்னும் மற்ற குறிகளாயினும் சோதனை 'பிட்' டையும் சேர்த்துக் கொண்டால் மொத்தம் எரிநிலையில் உள்ள 'பிட்'களின் எண்ணிக்கை ஒரு ஒற்றைப்படை எண்ணாக இருக்கும். படங்கள் 1-17, 1-18 ஆகியவற்றை பார்க்கவும்.

இந்த முறையைப் பயன்படுத்தி, ஏதாவது ஒரு குறியீடு நினைவகத்தின் ஓரிடத்திலிருந்து மற்றோரிடத்திற்குக் கொண்டு செல்லப்பட்டதும், கொண்டு செல்லப்பட்ட இடத்தில் ஒரு சோதனை நடக்கிறது—அதாவது வந்து சேர்ந்த 'பிட்'களில், எரிநிலையில் உள்ள 'பிட்'களின் எண்ணிக்கை ஒற்பைப்படை தானா என்று; அது ஒற்றைப்படை எண்ணாயிருந்தால் இட மாற்றத்தின் போது தவறு ஏதும் நிகழவில்லை என அறியப்படு கிறது. அதற்கு மாறாக இடப் பெயற்சியின் போது வழியில் ஒரு 'பிட்' கூடியிருந்தாலோ, அல்லது தவறியிருந்தாலோ எரிநிலை பிட்களின் எண்ணிக்கை இரட்டைப்படை எண்ணாக ஆகியிருக்கும். உடனே வழியில் ஏதோ தவறு நேர்ந்துள்ளது என அறியப்படுகிறது. இச்சோதனை முறையை 'ஒற்றைச் சமன் சோதனை' (Odd parity check) என்று அழைப்போம்.

மேற் சொன்ன சோதனை முறை முற்றும் நம்பத்தகுந்ததல்ல. ஏனெனில் இம்முறைப்படி ஒரே ஒரு 'பிட்' கூடியிருந்தால் **அல்லது** ஒரே ஒரு பிட் குறைந்திருந்தால் அத்தகையத் தவறை இச் சோதனை வெற்றிகரமாகக் கண்டுபிடிக்கும். ஆனால் ஒரே சமயத்தில் ஒரு பிட் இழக்கப்பட்டும், வேறு ஒரு பிட் புதிதாக வந்து சேர்ந்துமிருப்பின் அத்தவறை இச்சோதனை கண்டுபிடிக்க இயலாது.



படம் 1.15
'/.' குறிக்கான 'பிட்' அமைப்பு



படம் 1.16
'/.' குறியீட்டில் A-பிட் பழுதுற்று அணை நிலையில் இருத்தல்

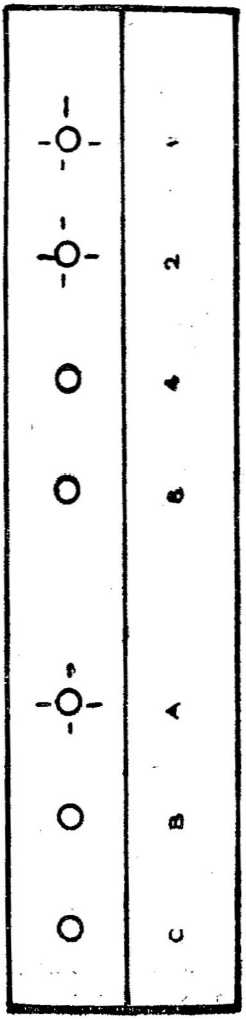
ஏனெனில் அப்போதும் மொத்த எரிநிலை பிட்டின் எண்ணிக்கை ஒற்றைப்படையாகவே இருக்கும். அவ்வாறே ஒரே சமயத்தில் இரண்டு பிட்களை இழந்தாலோ அல்லது இரண்டு 'பிட்'கள் புதிதாக வழியில் சேர்க்கப் பட்டிருந்தாலோ அப் போதும் இம் முறை பயன்படாது. ஆனால் ஒரே சமயத்தில் இரண்டு 'பிட்'களில் பிழை ஏற்படும் வாய்ப்பு மிகமிகக் குறைவு என்றும், நடை முறையில் அது பெரும்பாலும் ஏற்படாது என்றும் கணக்கிடப் பட்டிருக்கிறது. அதனால் நடை முறையில் மேற் சொன்ன 'ஒற்றைச் சமன் சோதனை' முற்றும் போதியதாகும்.

மேற்கூறிய சோதனை 'பிட்' மட்டுமின்றி, பிறிதொரு 'பிட்'டும் உள்ளது. இதனைக் 'கொடி' பிட் (flag bit) என்று அழைப் போம். இதன் பயனைப் மற்றோரிடத்தில் காண்போம். ஆக கொடி பிட்டையும் சேர்த்து நாம் இதுவரை கண்டபடி, மொத்தம் 8-பிட்களை (எண்பிட் $4+$பகுதி பிட் $2+$சோதனை பிட் $1+$ கொடி பிட் $1=8$) பயன்படுத்தி எண்களையும் எழுத்துக்களையும் மற்றக் குறியீடுகளையும் இருமை முறையால் நாம் குறிக்கிறோம். (படம் 1-19).

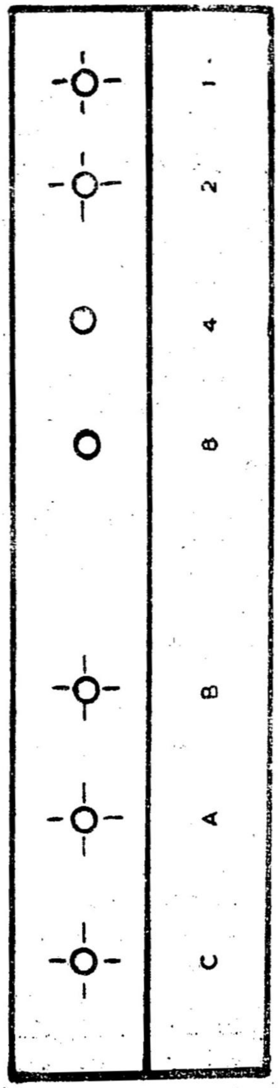
கம்ப்யூட்டரின் நினைவகம், முன் பகுதியில் குறிப்பிடப் பட்டுள்ள முறைகளில் ஒன்றைக் கொண்டுதான், கட்டளைகளையும், மற்ற விவரங்களையும் நினைவில் இருத்திக் கொள்ளுகிறது. மின் விளக்குகளுக்குப் பதிலாக ஆயிரக்கணக்கான மிகச்சிறிய மின் காந்த இரும்பு வளையங்கள் (Ferro Magnetic rings) உள்ளன. சதுர வடிவமான சட்டங்களில் குறுக்கும் நெடுக்குமாகச் செல்லும் மெல்லிய கம்பிகளின் சந்திப்பு முனைகளில் இவ்வளையங்கள் சேர்க்கப்பட்டுள்ளன. கம்பிகளில் மின்சாரத்தை செலுத்தியும், நிறுத்தியும் இம் மின் வளையங்களில் காந்த மேற்றியும் (Magnetised) காந்த மறுத்தும் (demagnetised), பைனரி முறையில் (இருமை முறையில்) எண்கள் குறிக்கப்படும்.

முதல் சட்டத்தில் உள்ள எல்லா மின் வளையங்களும் காந்த முற்ற நிலையில் (Magnetised state) 2^0 மதிப்பும், காந்தமற்ற நிலையில் O மதிப்பும் கொண்டவை. இதை '2^0-சட்டம்' என்போம்.

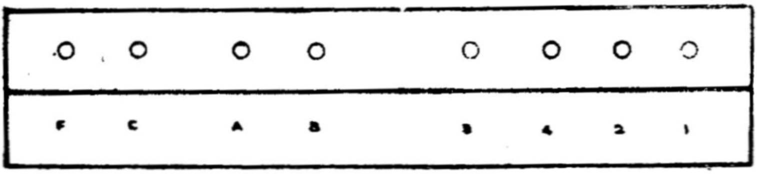
இரண்டாவது சட்டம் முதல் சட்டத்தை முற்றிலும் ஒத்து முதல் சட்டத்தின் பின்புறம் அமைக்கப்பட்டு இருக்கும். இரண்டாம் சட்டத்தின் ஒவ்வொரு மின்காந்த வளையத்திற்கும் காந்தமுற்ற நிலையில் 2^1 மதிப்பும் காந்தமற்ற நிலையில் O மதிப்பும் கொடுக்கப்படும்; இச்சட்டத்தை 2^1 சட்டம் என்று கூறுவோம்.



படம் 1.17
சோதனை 'பிட்' — அணை நிலையில்



படம் 1.18
சோதனை 'பிட்' — எரி நிலையில்



படம் 1.19

ஒரு நினைவறையின் முழுமையான 'பிட்' அமைப்பு

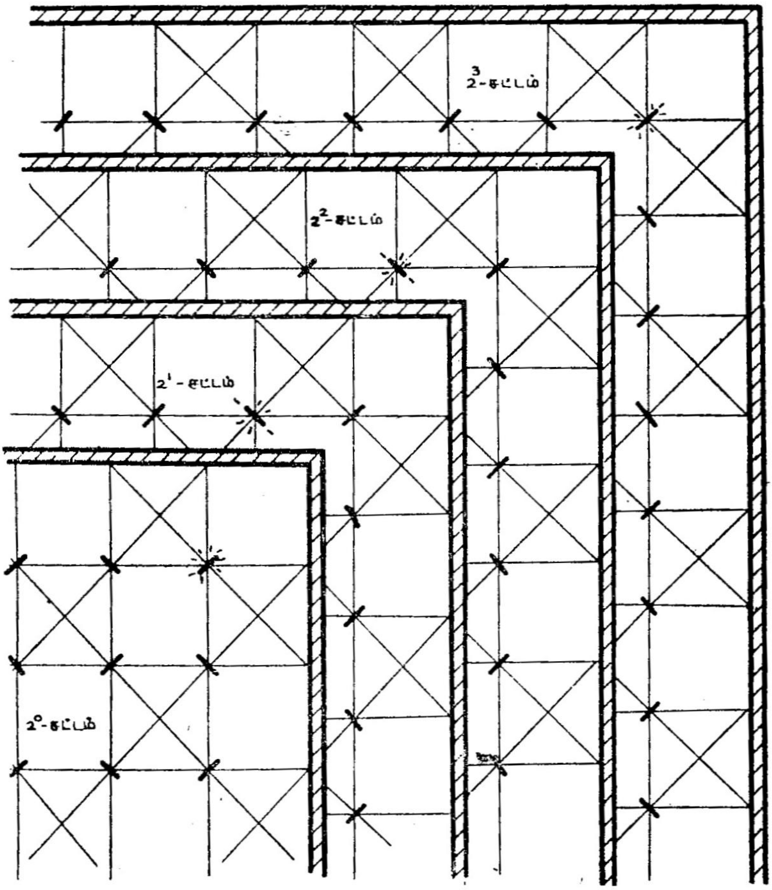
இவ்வாறே மூன்றாவது, நான்காவது சட்டங்களும் இரண்டாம் சட்டத்திற்கு பின்னால் முறையே ஒன்றன் பின் ஒன்றாக அமைக்கப் பட்டிருக்கும். இவ்வமைப்பின் படி, முதல் சட்டத்தில் உள்ள ஒவ்வொரு வளையமும் எரி நிலையில் 2^0 மதிப்புள்ளது. அதன் நேர் பின்புறம் உள்ள மற்ற ஏழு சட்டங்களில் முறையே 2^1, 2^2, 2^3, A, B, C, F பிட்கள் அமைந்திருக்கின்றன. (படம் 1-20).

நினைவறை

மேற்குறித்த அமைப்பின் படி முதல் சட்டத்தில் உள்ள காந்த வளையம், ஒவ்வொன்றும் அதன் நேர் பின்புறம் அமைந்துள்ள 7 வளையங்களுடன் சேர்ந்து ஒரு 8- வளையத் தொகுதியாக அமைந்துள்ளது. இத்தகைய 8- வளையத் தொகுதி ஒவ்வொன்றும் ஒரு எண்ணையோ, எழுத்தையோ அல்லது தனிக் குறியையோ (special character) இருமை முறைப்படி குறிக்க வல்லது. இத்தகைய 8- வளையத் தொகுதி ஒவ்வொன்றும் ஒரு நினைவறை (Memory cell) என்று அழைக்கப்படும்.

கம்ப்யூட்டரின் நினைவகத்தில் இத்தகைய நினைவறைகள் ஆயிரக்கணக்கில் உள்ளன. கம்ப்யூட்டரின் நினைவகத்தை ஒரு பெரிய, ஒரே அளவான, ஆயிரக் கணக்கான அறைகளைக் கொண்ட கட்டிடமாக உருவகித்தால், அக்கட்டிடத்தில் உள்ள அறை ஒவ்வொன்றையும் ஒவ்வொரு நினைவறையாக உருவகிக்கலாம். அம்மாபெரும் கட்டிடத்தில் உள்ள அறைகள் ஒவ்வொன்றிலும் ஒவ்வொரு மனிதன் இருப்பதாகக் கொள்வோம். இதற்கு இணையாக கம்ப்யூட்டரின் நினைவறை ஒவ்வொன்றிலும் உள்ள ஒவ்வொரு எண்ணையோ எழுத்தையோ, தனிக் குறியையோ கூறலாம். நமது சௌகரியத்தை முன்னிட்டு இனி நினைவகத்தின் அறைகளைக் கீழ்க்காட்டியவாறு கட்டங்களாக குறிப்போம். கட்டம் ஒவ்வொன்றும் ஒரு நினைவறையையும், அதனுள் காணப்படும் எண் (அல்லது எழுத்து அல்லது தனிக்குறி)

அந்த நினைவறையை உருவாக்கும் 8 பிட்களால் குறிக்கப் பெறும் எண்ணும் (அல்லது எழுத்து அல்லது தனிக்குறி) ஆகும். மேற்கூறியவாறு நினைவறையில் உள்ள எண்கள், கணிப்பின் போது மாறக்கூடும். எடுத்துக்காட்டாக (இடமிருந்து வலமாக மூன்றுவது அறையில் 'S' என்ற எழுத்து உள்ளது. கணிப்பின் போது இவ்வறையில் உள்ள 'S' அழிக்கப்பெற்று வேறு ஒரு எழுத்தோ அல்லது தனிக்குறியோ அவ்விடத்தை வந்து அடை



படம் 1.20

நினைவகத்தின் மாதிரி

(எரி நிலையில் காணும் மின்காந்த வளையங்கள் 69 என்ற எண்ணைக் குறிக்குமாறு அமைந்துள்ளதைக் காண்க)

யலாம். அல்லது இந்த 'S' வேறு ஒரு அறைக்கும் கொண்டு செல்லப்படலாம்) கணிப்பின்போது இவ்வாறு நினைவறைகளில் உள்ள எண்கள், எழுத்துக்கள் முதலியன இடம் மாறுதல் அல்லது அழிக்கப் பெறுதல் சர்வசாதாரணம்.

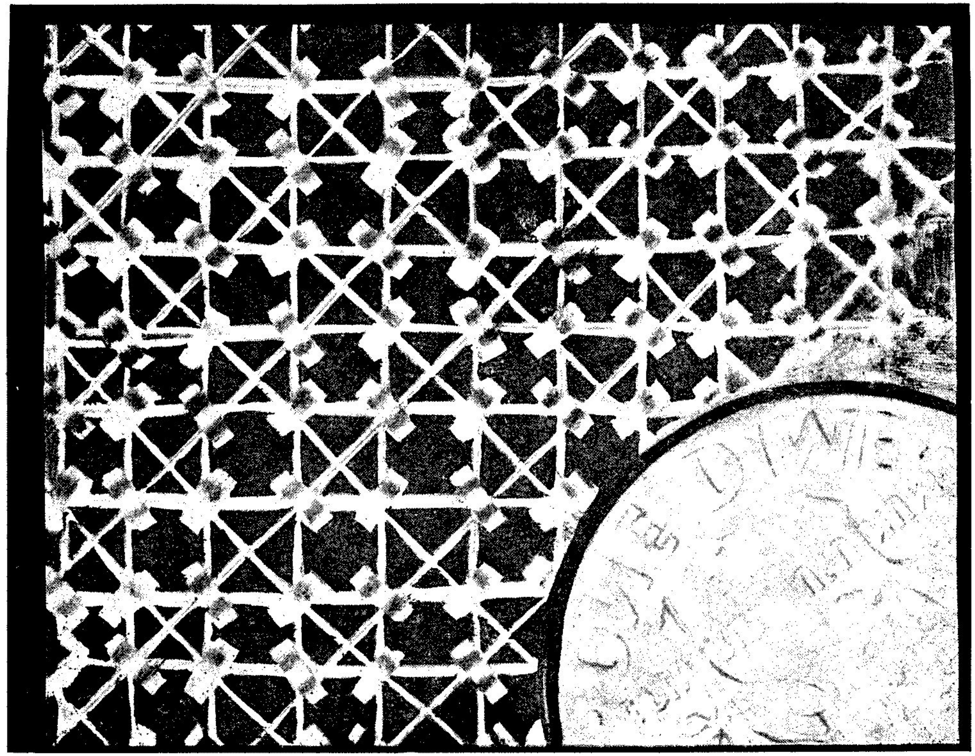
9	4	S	7	8

படம் 1.21

நினைவக அறைகளும் (உருவக அமைப்பு) அறைகளுள் உள்ள உருக்களும் (Characters)

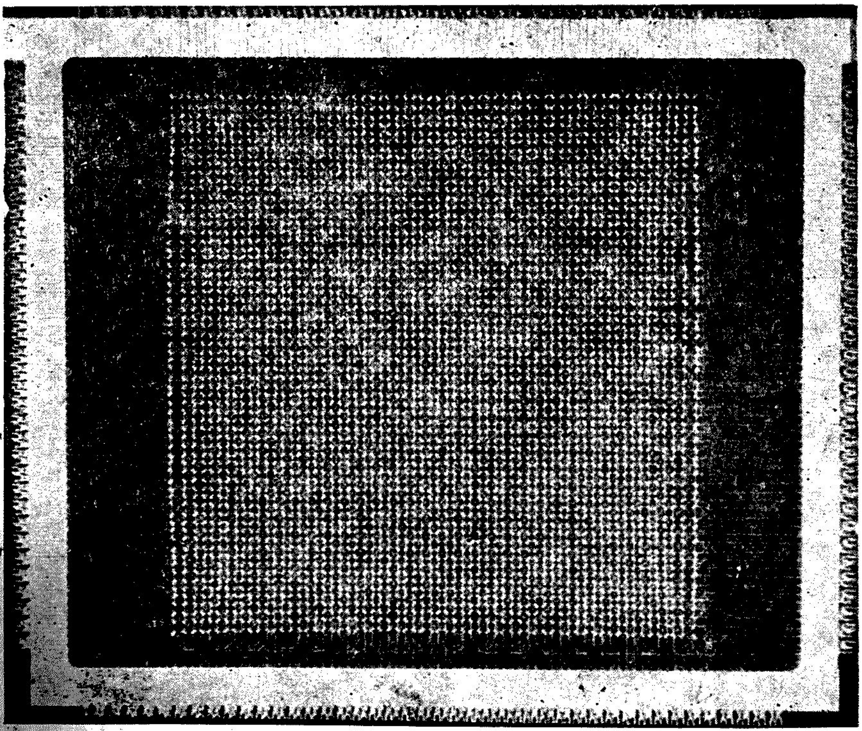
கணிப்பின்போது கம்ப்யூட்டரின் நினைவறைகளில் எண்கள், எழுத்துக்கள், தனிக் குறிகள் ஆகியவை நிரம்பி யிருக்கும். கட்டளைக் கொத்தின் கட்டளைகள், அடிப்படை விவரங்கள் ஆகிய யாவும் நினைவகந்தில்தான் இருக்கும். இவ்வாறு நிரம்பி இருக்கும் நினைவகத்திலிருந்து அவ்வப்போது ஒரு குறிப்பிட்ட எண்ணையோ, அல்லது வேறு விவரங்களையோ கம்ப்யூட்டர் அழைக்க வேண்டி யிருக்கும். இதன் பொருட்டு கம்ப்யூட்டரின் நினைவறை ஒவ்வொன்றுக்கும் ஒரு எண் அளிக்கப்பட்டிருக்கிறது— ஒரு ஊரில் உள்ள வீடுகளுக்கு வீட்டு எண் அளிக்கப்படுவது போல இந்த நினைவறை எண்களை அவ் வறைகளுக்கு உரிய 'முகவரி' (address) என்று கூறுவர். பெரும்பாலும் இந்த அறை எண்கள் 0-ல் ஆரம்பித்து வரிசையாக 0, 1, 2, 3 என்று கடைசி நினைவறை வரை செல்லும். இவ் வறைகளின் அறை எண்கள் மாற்றத் தக்கவை அல்ல, அவை கம்ப்யூட்டரின் வடிவமைப்பிலேயே உருவானவை. நினைவறைகளில் உள்ள எண்களையோ எழுத்துக்களையோ பெறவேண்டுமானால் கம்ப்யூட்டர் அந்த எண்களையோ எழுத்துக்களையோ நேரிடையாக அழைக்க இயலாது. அந்த எழுத்துக்கள் அல்லது எண்களின் முகவரியைப் பயன்படுத்தி தான் அம்முகவரியில் உள்ள எண்ணை எழுத்தைப் பெறவேண்டும்.

இது எவ்வாறு நடைபெறுகிறது என்று காண்போம். படம் 1.22-ல் நினைவக மாதிரி ஒன்று காட்டப்பட்டுள்ளது. இம் மாதிரியில் ஒவ்வொரு நினைவறையின் அடியில் சிறிய எழுத்தில் காணப்படும் மூன்றிலக்க எண்கள் அந்தந்த நினைவறையின்



படம் 1.20a

நினைவக வளையங்களின் பருமனை அருகிலுள்ள நாணயத்துடன் ஒப்பு நோக்குக



படம் 1.20b

ஒரு நினைவகச் சட்டத்தின் முழுத் தோற்றம்

				×	$	1	5	5	·
750	751	752	753	754	755	756	757	758	759
9	9	9	$\overline{F}$	$\overline{\perp}$	N	E			
760	761	762	763	764	765	766	767	768	769
		$\overline{D}$	A	T	E		2	/	7
770	771	772	773	774	775	776	777	778	779
/	7	2	$\overline{R}$	S	7	5	=	9	5
780	781	782	783	784	785	786	787	788	789
$\overline{M}$	4	7	9	$\overline{2}$	6	8	A	5	6
790	791	792	793	794	795	796	797	798	799
O	7	5	Θ	$\overline{B}$	3	5	0	M	1
800	801	802	803	804	805	806	807	808	809

படம் 1.22

நினைவகத்தின் மாதிரி (ஒரு பகுதி)

முகவரியாகும். சில எழுத்துக்களின் தலைமீது ஒரு '—' குறி காணப்படுகிறது. இந்தக் குறி அந்த நினைவறையைச் சேர்ந்த கொடி-பிட் (Flag bit) எரி நிலையில் இருக்கிறது என்பதைக் குறிக்கிறது. இக்குறி இல்லாத நினைவறைகளில் இந்த பிட் அணை-நிலையில் உள்ளதென்று அறியப்படும். கம்ப்யூட்டர் தனக்கு வேண்டிய விவரங்களை நினைவகத்திலிருந்துபெற விழை யும்போது உரிய நினைவறையை அதனுடைய முகவரியை விழிப் பதன் மூலம் பெறுகிறதென்று சற்று முன் கூறினோம். எடுத்துக் காட்டாக கம்ப்யூட்டர் முகவரி 775-ஐ விழிக்கிறதென்று வைத் துக்கொள்வோம். இப்போது முகவரி 775-ல் உள்ள E என்ற எழுத்து கிடைக்கிறது. அத்துடன் நில்லாமல் 775-ல் ஆரம் பித்து வலமிருந்து இடமாக அடுத்தத்து உள்ள நினைவறைகளில் உள்ளவற்றையும் தானாகவே கம்ப்யூட்டர் சேகரிக்கிறது. இவ் வாறு சேகரிக்கும்போது ஏதாவது ஒரு நினைவறையில் கொடி-பிட் எரிநிலை இருப்பின் அவ்வறையில் உள்ள எண்ணையோ, எழுத்தையோ சேகரித்த உடன் இச் சேகரம் நின்றுவிடும். முன் கூறியவாறு முகவரி 775-ஐ விழித்தவுடன் முதலில் E சேகரிக்கப்படுகிறது; பின்னர் T, அதன் பின்னர் A, அதற்கும் பின்னர் D ஆகிய எழுத்துக்கள் அவற்றின் இட அமைப்பு மாறுமல் சேகரித்துக்கொள்ளப்படுகின்றன. முகவரி 772-க்கு வந்த உடன் D என்ற எழுத்தின்மீது '—' குறி எதிர்படுகின் றது. (அதாவது D இருக்கும் நினைவறையில் கொடி-பிட் ஏற்றப் பட்டுள்ளது) ஆகவே D-யை சேகரித்த உடன் சேகரம் (collection) நின்று விடுகிறது. ஆக கம்ப்யூட்டர் DATE என்ற வார்த் தையைச் சேகரித்துக் கொள்ளுகிறது. ஒன்றை நாம் நினைவில் வைத்துக்கொள்ள வேண்டும்.

கம்ப்யூட்டர் விழிப்பது 'E' இருக்கும் முகவரியை மட்டுமே அதற்கு இடப்புறம் உள்ளவை யாவும் தாமாகவே சேகரிக்கப் படுகின்றன. இவ்வாறு சேகரிக்கப்பட்ட தொகுதியை நாம் 'தொடர்' (held) என்று அழைப்போம். இத் தொடருக்கு ஒரு முகவரியும் கொடுப்போம். ஒரு தொடரின் முகவரி, அத்தொட ரின் வலப்புறம் கடைசியாக உள்ள எண், எழுத்து அல்லது தனிக் குறியின் முகவரியாகும். படம் 1.19-ஐப் பயன்படுத்தி தயாரிக்கப்பட்ட அட்டவணை 1.3. சில தொடர்களையும் அவற் றின் முகவரிகளையும் காட்டுகிறது.

கம்ப்யூட்டர் செயல்படும்போது அதற்கான கட்டளைக்கொத் தின் (Program) கட்டளைகளில் நினைவறைகளின் முகவரிகள்மட் டுமே பயன்படுத்தப்படும். அந்த முகவரிகளை பயன்படுத்தித்

தான், கூட்டல், கழித்தல், பெருக்கல், வகுத்தல் முதலிய **ஆணைகள் (Instructions)** எழுதப்பட வேண்டும். எடுத்துக்காட்டிற்காக 'கழித்தல்' ஆணை எவ்விதம் செயல்படுகிறது என்பதைக் காண்போம்.

	தொடர்	தொடரின் முகவரி
1	FINE	766
2	R	783
3	26	795
4	RS 75 = 95	789
5	RS 75 =	787
6	M 479	793
7	DATE 2/7/72	782
8	M	790
9	DA	773
10	B 35 OM 1	809

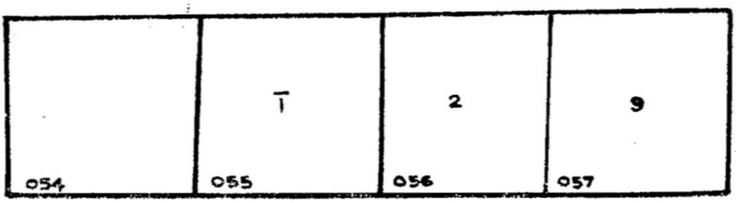
அட்டவணை 1.3

ஆணை

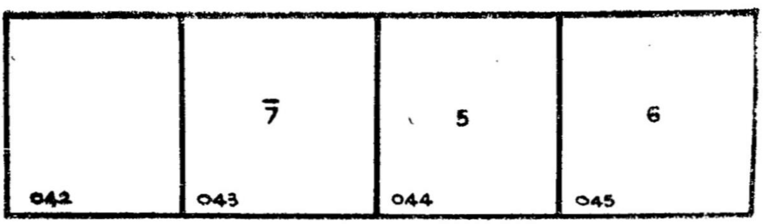
S 057 045

மேலுள்ள தொடரில் உள்ள S என்று எழுத்து 'கழிக்க' (Subtract) என்ற கட்டளையாகும். அடுத்து உள்ள இரண்டு மூன்றிலக்க எண்கள் இரண்டும் இரண்டு முகவரிகளாகும். எந்த ஒரு கட்டளையிலும் முதலாவதாக உள்ள முகவரியை A முகவரி என்றும் அடுத்துள்ளதை B முகவரி என்றும் குறிப்பர். ஆகவே மேலுள்ள கட்டளையில் ஒரு கட்டளைக்குறி (Operation code)யும் இரண்டு முகவரிகளும் ஆகியவை உள்ளன. மேலுள்ள கட்டளையை நாம் விரித்தெழுதினால் '057' முகவரியில் உள்ள தொடரை '0 45' முகவரியில் உள்ள தொடரிலிருந்து கழிக்கவும், என்று ஆகும். எடுத்துக் காட்டாக முகவரி 057 ல் 129

என்ற எண் தொடரும் முகவரி 045 ல் 756 என்ற எண் தொடரும் இருப்பதாகக் கொள்வோம். நினைவகத்தில் இவ்விரு தொடர்களும் கீழுள்ளவாறு காணும்.



படம் 1.23 a
முகவரி 057-ல் உள்ள தொடர்



படம் 1.23 b
முகவரி 045-ல் உள்ள எண் தொடர்

மேலுள்ள ஆணையைக் கம்ப்யூட்டர் நிறைவேற்றும் போது 057 என்ற முகவரியை விழித்து 129 என்ற எண்ணைப் பெறுகிறது. முகவரி 045ஐ விழித்து 756 என்ற எண்ணைப் பெறுகிறது. பிறகு 756 லிருந்து 129ஐக் கழிக்க ஆரம்பிக்கிறது.

கீழுள்ள அட்டவணை 1.4-ல் இன்னும் சில மாதிரி ஆணைகளும் அவற்றின் விளக்கங்களும் கொடுக்கப்பட்டுள்ளன.

ஆணைக்குறி	A-முகவரி	B-முகவரி	விளக்கம்
M	125	475	முகவரி 125-ல் உள்ள விவரத்தை 475-முகவரிக்கு எடுத்துச் செல். (M-Move என்பதன் சுருக்கம்)
1			ஒரு கார்டைப் படி
4			ஒரு கார்டைத் துளை
/	0 80		* முகவரி 000-விலிருந்து 080 வரையான நினைவகப் பரப்பைத் துடை

* இவ்வாணை ''கரும்பலகையைத் துடை'' என்பதைப் போன்றது.

அட்டவணை 1.4

2
கொள் கொடு துறைகள்

(Input/Output/Media)

சென்ற அத்தியாயத்தில் மனிதர்களைப் போலவே கம்ப் யூட்டர்களும் கட்டளைகளை ஏற்று, நினைவில் கொண்டு பின்பு அவற்றை நிறைவேற்றவல்லன என்று பார்த்தோம். கம்ப் யூட்டர்களுக்குக் கட்டளைகளை எவ்வாறு கொடுப்பது? மனிதர் களாயின் நாம் அவர்களிடம் வாய்ப்பேச்சு மூலமாகவோ அல்லது காகிதத்தில் எழுதிக் காட்டியோ இத்தகைய ஆணை களை இடமுடியும். ஆனால் இயந்திரத்திற்கு இவ்வாறு ஆணை யிடுவது என்பது இயலாத ஒன்று. மனிதர்களிடம் பேசுவது போல் இயந்திரத்தின் முன்னால் நின்று நமது கட்டளைகளை வாய்விட்டுக் கூறவோ அல்லது காகிதத்தில் எழுதிக்காட்டவோ முடியாது. மனிதர்களின் கையெழுத்தையோ, வாய்வார்த்தை களையோ நேரிடையாக ஏற்றுப் புரிந்து கொள்ளவல்ல கருவிகள் இன்னும் ஏற்படவில்லை. இத்தகைய கருவிகளை வடிவமைப்பதில் உள்ள முக்கியமானதும் கடினமானதுமான பிரச்சினை, மனிதர் களின் வார்த்தை உச்சரிப்பில் உள்ள வேறுபாடேயாகும். எல்லா மனிதர்களும் எந்த ஒரு வார்த்தையையும் எப்போதும் ஒரே மாதிரி உச்சரிப்பதோ எழுதுவதோ இல்லை. பார்க்கப் போனால் ஒரே மனிதன் கூட ஒரு வார்த்தையை இரண்டு முறை உச்சரிக்கும் போது அதில் வேறுபாடு இருக்கிறது. இக் காரணத்தால் தான் இதுவரை வாய்வாாத்தைகளைக் 'கேட்டு' புரிந்து கொள்ளக் கூடிய இயந்திரங்கள் வடிவமைத்துத் தயா ரிக்க இயலவில்லை. ஆயினும் இந்தத் துறையில் (Speech recognition) பலவேறு ஆராய்ச்சிகள் இன்று நடந்து வரு கின்றன. இவ்வாராய்ச்சிகளின் நோக்கம் வார்த்தைகளின் உச்சரிப்பில் உள்ள வேறுபாடுகளைக் களைந்து உரிய வார்த்தை களை அடையாளம் கண்டுகொள்ள-வல்ல பொறி ஒன்றைக் கண்டுபிடிப்பதே. இத்தகைய பொறி ஒன்று வடிவமைக்கப்

பட்டு செயல்படுமானல் நாம் அவற்றுடன் பேசமுடியும். நமது வாய் வார்த்தை அல்லது எழுத்து மூலம் கம்ப்யூட்டரும் அதற்கான ஆணைகளைப் பெற முடியும். ஆனல் இன்றைய நிலையில் உச்சரிப்பு வேறுபாடுகளாலோ எழுத்து வேறுபாடுகளாலோ பாதிக்கப்படாத ஒரு அமைப்பைக் கொண்டுதான் கம்ப்யூட்டர்களுக்கு நாம் கட்டளைகளையும் அடிப்படை விவரங்களையும் (Data) கொடுக்க வேண்டியிருக்கிறது. இத்தகைய அமைப்புகளில் முக்கியமானவை காகிதக் கார்டும் (Card) காகித நாடா (Paper tape)வும் ஆகும். இவற்றைப் பற்றி சிறிது விளக்கமாகப் பார்ப்போம்.

கார்டுக் குறி. (Card—Code).

கம்ப்யூட்டரில் பயன்படுத்தப்படும் கார்டுகளில், குறிப்பிட்ட இடங்களில் துளைத்து, இத்துளைகளை ஒரு குறியீடாகக் (Code) கொண்டு நமது கட்டளைகளும் இன்னபிறவும் கம்ப்யூட்டருக்கு அளிக்கப் படுகின்றன. முதலில் 'கார்டுக்குறி'யைப் பற்றி விளக்கமாகப் பார்ப்போம். அதன் பின்னர் கார்டின் உபயோகத்தைத் தெரிந்து கொள்ளுவது எளிதாக இருக்கும்.

கார்டுக் குறி மாதிரிகள் :

படம் 2-1ல் நெடுவாக்கில் காட்டியிருப்பது ஒரு அட்டைத் துண்டு. இவ்வட்டைத்துண்டில் 0,1,2,3...9 ஆகிய எண்கள் ஒரே சீரான இடைவெளிகளில் ஒன்றின்கீழ் ஒன்ருக அச்சிடப் பட்டுள்ளன. 0-க்கு மேலே சிறிது இடைவெளி விடப்பட்டுள்ளது. இவ்விடைவெளியில் 12,11 என்ற இலக்கங்கள் காணப்படுகின்றன. ஆக இந்தக் காகிதத் துண்டில் மொத்தம் 12 இடங்கள் குறிப்பிடப் பட்டுள்ளன. இப்போது, இக்காகிதத் துண்டின் அமைப்பைக் கொண்டு ஒரு புதிய குறியீட்டை ஏற்படுத்துவோம். ஒரு காகிதத் துண்டின் ஒரே ஒரு எண் அல்லது எழுத்து அல்லது சிறப்புக் குறி மட்டுமே குறியிடப் படும். ஒவ்வொரு எழுத்து அல்லது எண் அல்லது சிறப்புக் குறிக்கும் ஒரு காகிதத் துண்டு பயன்படுத்தப்பட வேண்டும். குறிமுறை (Coding procedure) பின்வருமாறு :- எந்த இலக்கத்தை (0 முதல் 9 வரை) குறிக்க விரும்புகிருேமோ அந்த இலக்கத்தை இக்காகிதத்துண்டிலிருந்து துளைத்து எடுத்துவிட வேண்டும். எடுத்துக் காட்டாக 5 என்ற எண் இருக்குமிடத்தைத் துளைத்து எடுத்துவிட்டால் இக்காகிதத் துண்டு 5 என்ற இலக்கத்தைக் குறிப்பிடும். படம் 2-1 bல் இவ்வமைப்பு காட்டப்பட்டுள்ளது. நீள் சதுரவடிவத்தில் உள்ள கருநிழல் துளைத்தெடுக்கப்பட்ட இடத்தைக் குறிக்கிறது.

A முதல் Z வரை உள்ள எழுத்துகளைக் குறிக்க இரண்டு துளைகள் போடுவோம். ஒரு துளை இலக்கத்தின் மீதும் மற்றது 12 அல்லது 11 அல்லது 0 மீதும் இருக்கும். எழுத்திற்கு ஏற்றவாறு இலங்கங்களில் ஒன்றும், 12, 11 அல்லது 0-வில் ஒன்றும் துளைத்தெடுக்கப்படும். எடுத்துக் காட்டாக 2.1 C யில் A என்ற எழுத்தைக் குறிக்க 12 இடமும் 1 என்ற இடமும் துளைத்தெடுக்கப் பட்டிருக்கின்றன. அட்டவணை 2-1ல் இலக்கங்கள், எழுத்துக்கள் மற்றும் சிறப்புக் குறியீடுகள் ஆகியவற்றின் துளைக்குறிகள் முற்றுமாக கொடுக்கப் பட்டுள்ளன.

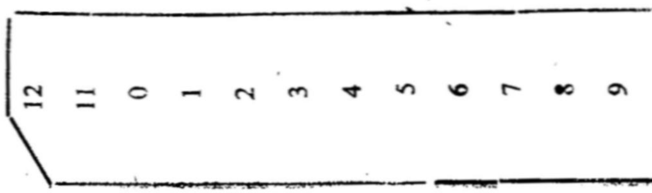
படம் 2-1a முதல் 2-1C முடிய ஒவ்வொன்றும் ஒரு காகிதத் துண்டு என்று கூறினோம். இதன்படி ஒரு சொல்லைக் குறிப்பிட அச்சொல்லில் எத்தனை எழுத்துக்கள் உண்டோ அத்தனைத் துண்டுகள் தேவைப்படும். அதற்கு மாறாக பெரிய காகிதத் துண்டில், முன் குறித்த மாதிரி எண் வரிசைகளை அருகருகே அச்சிட்டால் ஒவ்வொரு நெட்டு வரிசையையும் ஒவ்வொரு எண் அல்லது எழுத்தைக் குறிக்கப் பயன்படுத்திக் கொள்ளலாம். படம் 2-2ஐப் பார்க்கவும். தற்போது கம்ப்யூட்டரில் பயன் படுத்தப்படும் கார்டில் 80 நெட்டு வரிசைகள் உள்ளன.

படம் 2-2ல் உள்ள பெரிய காகிதத் துண்டைக் கொண்டு 'JAN-15' என்பதைத் துளைக்குறி மூலம் குறித்தால் படம் 2-3ல் உள்ளதைப் போல் இருக்கும். (படம் 2-3ல் 12; 11 ஆகிய எண்கள் எழுதப்படவில்லை என்பதை நோக்குக.)

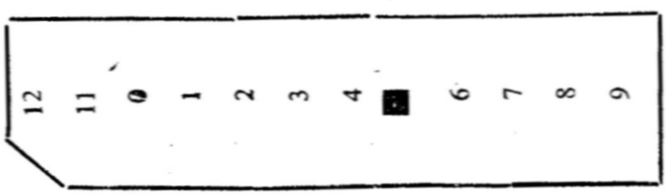
மேலே கண்ட முறையில்தான் கம்ப்யூட்டர் கார்டில் துளைக் குறி முறையில் செய்திகள் குறிக்கப்படுகின்றன. இவைபற்றி அடுத்த பகுதியில் பார்ப்போம்.

கார்டு (The Card)

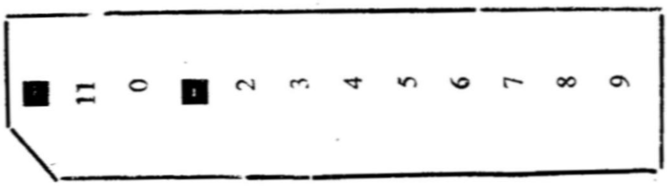
கம்ப்யூட்டர் உலகில் இந்தக் காகிதக் கார்டு இன்றியமையாதது. இக்கார்டுகளைக் கொண்டுதான் நமது கட்டளைகளும் அடிப்படை விவரங்களும் கம்ப்யூட்டருக்குச் சுமந்து செல்லப் படுகின்றன. பார்க்கப் போனால் இக்காகித அட்டைகளின் வரலாறு கம்ப்யூட்டரின் வரலாற்றைவிட பழமையானது. இன்றைய கம்ப்யூட்டர்கள் வருவதற்கு வெகு நாட்களுக்கு முன்னரே அட்டைகளைக் கொண்டு இயந்திரங்களின் இயக்கத்தை வழி நடத்தும் முறை கண்டுபிடிக்கப்பட்டது. பத்தொன்பதாம் நூற்றாண்டின் தொடக்கத்தில் (1801) "ஜோசப் மாரி ஜெக்கார்ட்" என்ற பிரெஞ்சுக்காரர் காகித அட்டைகளில் ஏற்ற துளைகளை



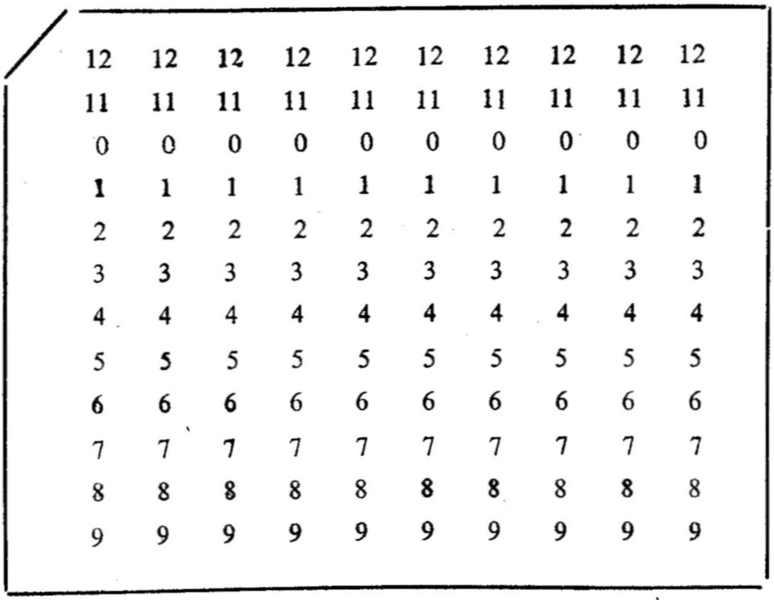
படம் 2.1 a



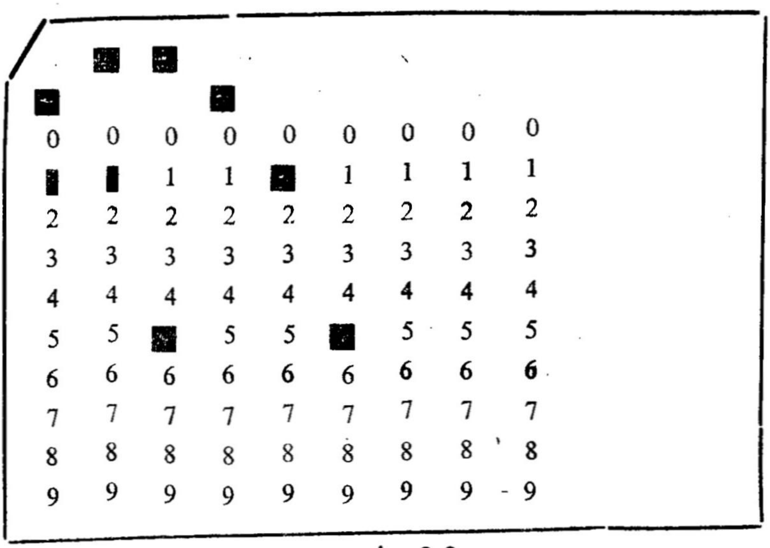
படம் 2.1 b
'5' என்ற எண்ணிற்கான துளைக்குறியீடு



படம் 2.1 c
'A' என்ற எழுத்துக்கான துளைக்குறியீடு

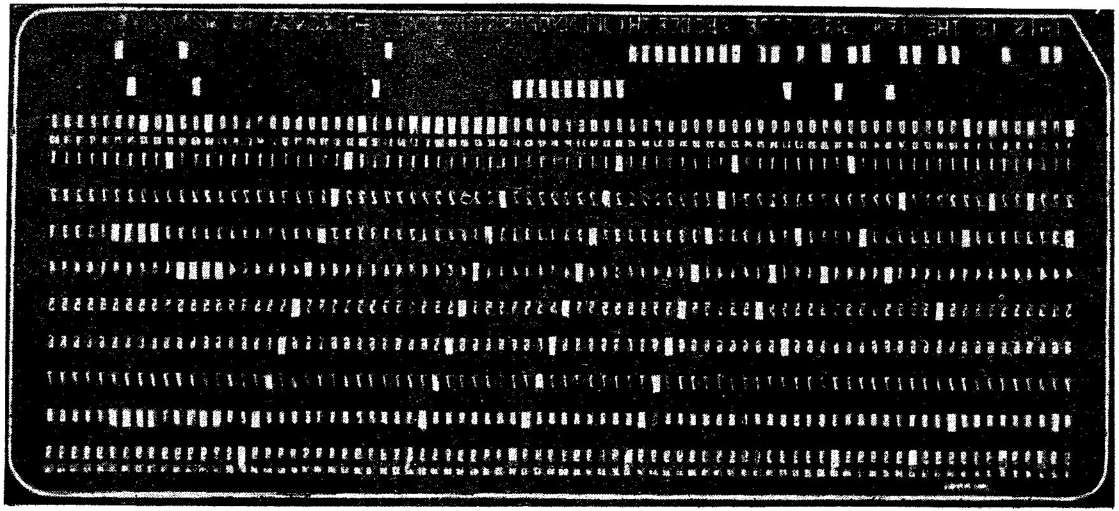


படம் 2.2



படம் 2.3

'JAN — 15' என்ற தொடருக்கான துளைகள்



படம் 2.4

1BM துளை அட்டை

அமைத்துத் துணி நெய்யும் தறிகளில் பயன்படுத்தினர். இத் துளையட்டைகள் நெய்யப்பட்ட துணிகளில் பலவேறு விதமான அழகிய வேலைப்பாடுகளை (Design) அமைத்தன. இந்த அழகிய வேலைப்பாடும் ஜெக்கார்டின் கண்டுபிடிப்பும் அன்றைய பிரெஞ்சு மக்களை மிகவும் கவர்ந்தன. அதன் பயனாக பிரான்ஸ் நாட்டிலேயே மிக உயர்ந்த மரியாதைப் பட்டமான 'லீஜன் ஆப் ஆனர்' என்ற பட்டம் பிரெஞ்சு அரசாங்கத்தால் ஜெக்கார்டிற்கு அளிக்கப்பட்டது.

இன்றைய கம்ப்யூட்டரின் தந்தை என்ற கூறத்தகுந்தவர் 'சார்லஸ் பாபேஜ்' என்ற ஆங்கிலேயர். கணிதத்துறை வல்லுன ரான பாபேஜ் (கம்ப்யூட்டர் அறிமுகமும் வரலாறும் என்னும் தலைப்பின் கீழ் பார்க்க) ஜோஸப் ஜெக் கார்டின் துளை அட்டை யின் அடிப்படையைப் பயன்படுத்தியேகட்டளைக் கொத்தேற்க வல்ல (Stored Program) கம்ப்யூட்டர் (பாபேஜ் இதை அனலடிக் என்ஜின் 'Analytic Engine' என்று அழைத்தார்) ஒன்றை வடிவமைப்பதில் முனைந்தார். ஆனால் பாபேஜ் இந்த இயந் திரத்தை செய்து முடிக்காமலேயே 1871-ம் ஆண்டு இறந்தார் என்று முன்னரே பார்த்தோம்.

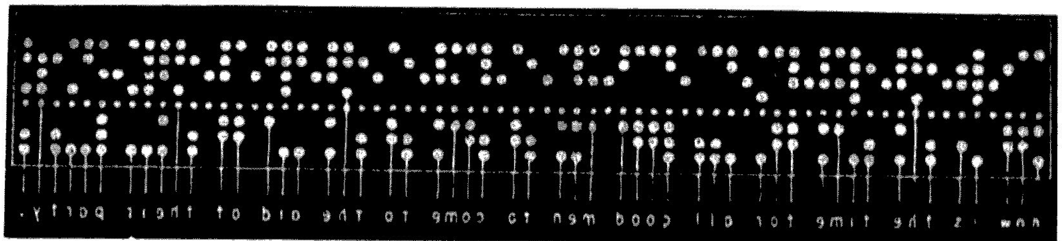
ஜோசப் ஜெக்கார்டின் துளை அட்டையைப் பயன்படுத்திய அடுத்த நபர், ஹெர்மன் ஹாலரித் (Herman Hallarith) என்ற அமெரிக்கர் ஆவார். இவர் 19-ஆம் நூற்றாண்டின் கடைப் பகுதியில் (1880) ஐக்கிய அமெரிக்காவின் (U.S.A.) மக்கள் தொகைக் கணிப்புத் துறையின் அதிகாரியாயிருந்தவர். அவர் காலத்திலேயே அமெரிக்க ஐக்கிய நாட்டின் பார்லிமெண்ட், தமது நாட்டில் பத்து ஆண்டுகளுக்கு ஒருமுறை மக்கள் தொகைக் கணிப்பு மேற்கொள்ளப் பட வேண்டும் என்று சட்டம் இயற்றியது. அவ்வாறு கணிக்கப்பட்ட மக்கள் தொகை பற்றிய புள்ளி விவரங்களை அந்நாட்டின் புள்ளி விவரத் துறை தொகுத்து ஆராய்ந்து, பயனுள்ள முடிவுகளைக் காண 12 ஆண்டுகள் ஆயின. இக்கால தாமதத்தால், தொகுக்கப்பட்ட புள்ளி விவரத்தின் பயன் ஏதுமின்றிப் போயிற்று. அதனால் அன்றைய அமெரிக்க அரசாங்கம் இப்புள்ளி விவரங்களை அட்ட வணையிட்டு (tabulate) ஆயும் பணியை இயந்திர மயமாக்கும் வழிகளை ஆயும்படி ஹாலரித்தைக் கேட்டுக் கொண்டது. அதன் பயனாக, ஹாலரித் சில இயந்திரங்களை வடிவமைத்துத் தயாரித்தார். இவ்வியந்திரங்கள் 'யூனிட் ரெக்கார்ட் மெஷின்' (Unit Record Machines) என்று அமைக்கப் பட்டன. இவ்வி யந்திரங்கள், மக்கள் தொகைக் கணிப்புத் துறை மட்டு மில்லாமல் அமெரிக்காவின் பலவேறு வியாபார நிறுவனங்களின்

கவனத்தையும் கவர்ந்தன. தனது இயந்திரங்களுக்குப் பெரும் வரவேற்பு இருப்பதைக் கண்ட ஹாலரித் தாம் பார்த்த அரசாங்க வேலையை விட்டுவிட்டு, தனது இயந்திரங்களைத் தயாரிக்க ஒரு கம்பெனியை ஏற்படுத்தினர். அந்தக் கம்பெனியே, பலவித மாற்றங்களுக்குப் பின்னர் இன்றுள்ள IBM (International Business Machines) என்ற அமெரிக்கக் கம்ப்யூட்டர் கம்பெனியாகும். ஹாலரித்திற்குப் பிறகு அவர் பதவியை வகித்தவர் ''ஜேம்ஸ் பவர்ஸ்'' (James Powers) என்பவர் ஆகும். அவரும் ஹாலரித்தைப் போலவே, சில இயந்திரங்களை வடிவமைத்துத் தயாரித்தார். அவரும் தமது அரசுப் பதவியை உதறிவிட்டுத் தானும் தனது பொறிகளைத் தயாரித்து விற்பதற்கென ஒரு கம்பெனியை ஆரம்பித்தார்.

ஹாலரித், தமது இயந்திரங்களில் பயன்படுத்துவதற்காக, ஜோசப் மேரி ஜெக்கார்டைப் பின்பற்றி காகித அட்டைகளைத் தயாரித்தார். அந்த அட்டைகளின் நீள அகலங்கள் அன்றைய ஒரு டாலர் தாளின் (One dollar Bill) நீள அகலங்களுடன் தயாரிக்கப்பட்டன. இன்றைய கம்ப்யூட்டர்களில் பயன்படுத்தப்படும் அட்டைகளில் பெரும்பாலானவைகளும் ஹாலரித் அட்டையின் அடிப்படையிலேயே அமைந்துள்ளன. வேறு சில அளவுகளில் அட்டைகள் தயாரிக்கப்படினும் பெரும்பாலானவை ஹாலரித்தின் அளவுகளையே கொண்டுள்ளன. ஹாலரித் கார்டின் அளவு $7\frac{3}{8}''$ நீளமும் $3\frac{1}{4}''$ அகலமும் கொண்டதாகும்.

ஹாலரித் அட்டையில் (இது IBM அட்டை என்றும் அழைக்கப்படும்) நாம் குறிப்பிட விரும்பும் விவரங்கள் யாவும் துளைவடிவில் குறியிடப் பெறுகின்றன. கீழுள்ள படத்தில் (படம் 2-4) இத்தகைய அட்டை ஒன்று காட்டப்பட்டுள்ளது.

படம் 2-4ல் காட்டப்பட்டுள்ள IBM அட்டைக்கும் படம் 2-2ல் உள்ள காகிதத்துண்டுக்கும் அமைப்பில் வேறுபாடு ஒன்றுமில்லை. படம் 2-2ல், 10 நெட்டுவரிசைகள் (columns) உள்ளன. 12, 11 என்ற எண்கள் அச்சிடப்பட்டுள்ளன. ஆனல் படம் 2.4-ல் உள்ள கார்டில் 80 நெடுவரிசைகள் உள்ளன. இந்நெடுவரிசைகளின் எண்ணிக்கைகள், கார்டின் கீழ்ப்புறத்திலும், 0-வரிசை, 1-வது வரிசை இவற்றிற்கு இடையிலும் அச்சிடப்பட்டிருப்பதைக் காணலாம். மேலும் படம் 2-2-ல் '11, 12' என்று அச்சிடப்பட்டுள்ள இடங்கள் கார்டில் வெற்றிடமாக இருப்பதைக் காணலாம். இவ்விரண்டு இடங்களும் அச்சிடப்படாவிடினும் அவ்விடங்களில் துளைகள் இடப்படும் 12, 11, 0 ஆகிய மூன்று



படம் 2.6

7 நாரைக் காகித நாடாத் துளைக் குறியீடுகள்

வரிசைகளைத் தொகுதி வரிசைகள் (zone rows) என்றும் 1,2,3...9 வரிசைகளை எண் வரிசைகள் (numeris rows) என்றும் கூறுவர்.

கம்ப்யூட்டரைப் பொருத்தவரை கார்டில் காணும் அச்சிடப் படும் விவரங்கள் பொருளற்றவை. ஏனெனில் கம்ப்யூட்டர்கள் கார்டில் உள்ள 'துளைகளைப் படித்துதான், அதிலுள்ள செய்தியை அறிகிறதே தவிர அதன் மீது அச்சிடப்பட்டுள்ள விவரங்களை அது படிப்பதில்லை. கார்டின் மீது அச்சிடப்பட்டுள்ளவை யாவும் மனிதர்களின் சௌகரியத்திற்காகவே. கார்டுகளில் துளைகளைத் தயாரிக்கும் போது சரியான இடத்தில் துளைகள் இடப் பட்டிருக்கின்றனவா என்று நாம் அறிந்து கொள்ளுவதற்காகவே 0; 1, 2...9 முதலிய இலக்கங்கள் அச்சிடப்பட்டுள்ளன. வேறு சில காரணங்களுக்காகவும் கார்டுகள் மீது வேறு சில செய்தி களும்— கார்டு பயன்படுத்தப்படும் சூழ்நிலையைப் பொருத்து அச்சிடப்பட்டிருக்கும். இவற்றைப்பற்றி பின்னர் பார்ப்போம்.

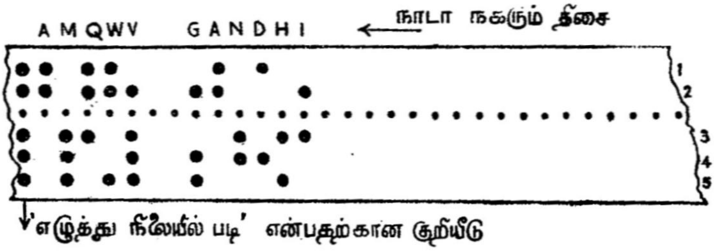
காகித நாடா (PAPER TAPE)

கம்ப்யூட்டருக்கு நாம் அளிக்க வேண்டிய கட்டளைகளும், அடிப்படை விவரங்களும் கார்டுகளில் துளைக்குறியிட்டு அளிக் கிறோம் என்று கூறினோம். இக்கார்டுகளைப் போலவே பயன் படும் மற்றொரு சாதனம் காகிதநாடா. கார்டுகளைப் போல காகித நாடா பெரு மளவில் இன்று கம்ப்யூட்டர் உலகில் பயன் படுத்தப் படுவதில்லை. அதுவுமல்லாமல் கம்ப்யூட்டர் உலகில் காகிதநாடா, சிறிது சிறிதாக புழக்கத்தில் இருந்து மறைந்து கொண்டே வருகிறது. இருப்பினும் இதைப் பற்றியும் நாம் சிறிது தெரிந்து கொள்ளுவோம்.

கார்டில் உள்ளதைப் போலவே, காகித நாடாவிலும் செய்தி களைத் துளைக்குறிகள் மூலமே கம்ப்யூட்டருக்கு அனுப்புகிறார்கள். ஆனால் கார்டின் துளைக்குறி முறைக்கும் காகித நாடாவின் துளைக் குறி முறைக்கும் வேறுபாடுண்டு.

படம் 2.5-ல் ஒரு காகித நாடா (துளைகளுடன்) காட்டப் பட்டுள்ளது.

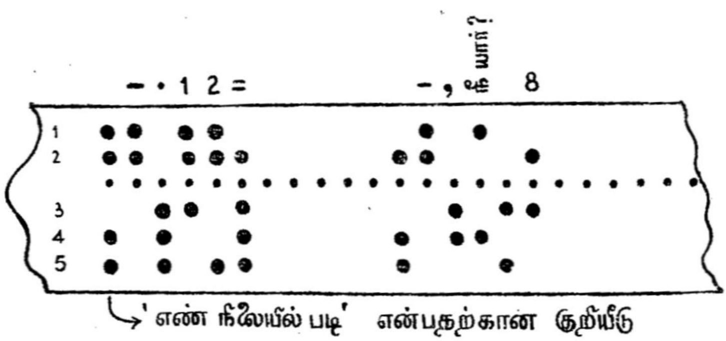
நாடாவின் மத்தியில் தொடர்ச்சியாக உள்ள துளைகள் நாடாவை ஒழுங்காகச் சுற்றிக் கொள்ளுவதற்காக அமைந்த பற்சக்கரங்களுக்கென அமைந்திருக்கின்றன. இவைதவிர, நாடா வில் காணும் மற்ற துளைகளே குறியீடுகளாகும்.



படம் 2.6

ஐந்து தாரை (tract) காகித நாடா

கார்டில் உள்ளதைப் போலவே காகித நாடாவிலும் நெடுக்காக (vertical position) உரிய துளைகளை இடுவதன் மூலம் எண்களும் எழுத்துக்களும் குறிக்கப் பெறுகின்றன. இவ்வாறு நெடுக்கில் துளையிடுவதன் பொருட்டு நாடாவின் நீளவாக்கில் 5 தாரைகள் (tracks) அமைந்துள்ளன.



படம் 2.7

W, V ஆகிய இரண்டு எழுத்துக்களைக் குறிப்பதற்கான துளைக் குறிகளைக் காணலாம். மேற்கூறிய 5-தாரை காகித நாடாக்களில் உள்ள ஒரு முக்கியமான அம்சம், ஒரே துளைச் சேர்க்கை (combination) இரண்டு வகைகளில் பயன்படுத்தப்படுவதாகும். எடுத்துக் காட்டாக படம் 2-6ல் 'W' என்ற எழுத்தைக் குறிக்க 1, 2, 5 தாரைகளில் துளையிடப்பட்டுள்ளது. இதே துளைச்சேர்க்கை "2" என்ற எண்ணையும் குறிக்குமாறு அமைந்துள்ளது. அதே போல் 2,3,4,5 தாரைகளில் துளைகள் இடப்படின் 'V' என்ற எழுத்தையும் '=' என்ற "ஈடுக் குறியையும்" குறிக்கிறது. இவ்வாறு ஒரே துளைச்சேர்கை இரண்டு விதமாகப் பயன்

படுத்தப் பட்டால் குழப்பம் ஏற்படாதா என வினவலாம். அன்றியும் கம்ப்யூட்டர் எப்படி ஒரு குறிப்பிட்ட துளைச் சேர்க்கையைக் காணும்போது அது இரண்டு விதங்களில் ஒன்றை இனம் பிரித்துக் காண்கிறது என்ற கேள்வியும் எழுகிறது.

சாதாரணமாக நாம் பயன் படுத்தும் ஆங்கில டைப்ரைட்டரில் எழுத்துக்கள் இரண்டு விதங்களில் (பெரிய எழுத்து, சிறிய எழுத்து) உள்ளன. சிறிய எழுத்துக்களும் (Small Case letters) மற்றும் சில குறிகளும் கீழ்நிலை (Lower key) யிலும், பெரிய எழுத்துக்களும் (Capitals) இன்னும் பிற குறிகளும் மேல் நிலையிலும் (Upper key) உள்ளன. ஒரு குறிப்பிட்ட பித்தானை, அதாவது மாற்றுப் பித்தானை (Shift key) அழுந்திய நிலையில் டைப் செய்யும் போது மேல் நிலை எழுத்துக்களும் அதை அழுத்தாமல் அடிக்கும் போது கீழ்நிலை எழுத்துக்களும் பொறிக்கப்படுகின்றன. இதைபோன்ற ஒருமுறையே மேற் சொன்ன காகித நாடாவில் பயன் படுகின்றது.

காகித நாடாவில் டைப்ரைட்டரில் உள்ளதைப் போல பித்தான் எதுவும் பயன்படுத்தப் படுவதில்லை. அதற்கு மாறாக இரண்டு துளைக்குறியீடுகள் பயன்படுத்தப் படுகின்றன. ஒரு குறிப்பிட்ட இடத்தில் 5 தாரைகளிலும் துளையிருக்குமானால் அதற்கு அடுத்து வரும் துளைச் சேர்க்கைகளை எழுத்து நிலையில் படிக்க வேண்டும் என்று பொருள். அதே போல் ஒரு குறிப்பிட்ட இடத்தில் 1,2,4,5 தாரைகளில் துளையிருப்பின் அக் குறிக்குப் பின் வருவனவற்றை எண் நிலையில் படிக்க வேண்டும் என்று பொருள். படங்கள் 2.6, 2.7ஐக் காண்க.

படம் 2-6ல் காகித நாடா வலமிருந்து இடமாகப் போகிறது. படிக்குமிடம் இடது புறத்தில் உள்ளது. இதன்படி முதலில் 1,2,3,4,5 தாரைகளில் துளைகளுள்ள ஒரு துளைச்சேர்க்கை எதிர்ப்படுகறது. இதன் பொருள் இனிவரும் துளைச் சேர்க்கைகளை எழுத்து நிலையில் படிக்கவே வண்டும் என்பதாம். இந்நிலையில் அடுத்துவரும் துளைத் தொகுதிகள் (AMQWV GANDHI) என்ற எழுத்துக்களைக் குறிக்கின்றன. முதலில் வரும் 1,2,3, 4,5 துளைச் சேர்க்கைக்கு மாறாக 1,2,4,5 துளைச் சேர்க்கை இருப்பின் அதற்குப்பின் வரும் துளைச் சேர்க்கைகள் எண் நிலையில் என்று படிக்கப்பெறும். படம் 2.7-ல் உள்ளதைக் காண்க. படம் 2.6-ல் :காணும் துளைச் சேர்க்கைகளும் 2.7-ல் காண்பனவும் ஒன்றே என்பதையும் ஆனால் படிப்பு நிலைக்கு

(எண்/எழுத்து நிலைகள்) ஏற்ப அவை குறிக்கும் உருக்கள் மாறுபடுகின்றன என்பதையும் காண்க.

இத்தகைய ஐந்து தாரை காகித நாடாக்களைப் பயன் படுத்தும் போது நாம் கவனமாக இருக்க வேண்டும். சாதாரண மாக நாம் கம்ப்யூட்டருக்கு அளிக்கும் செய்திகள் எண், எழுத்து, சிறப்புக்குறிகள் யாவும் கலந்தவையாயிருக்கும். எண் நிலை யிலிருந்து எழுத்து நிலைக்கு மாறும் போதும், எழுத்து நிலை யிலிருந்து எண் நிலைக்கு மாறும் போதும் கவனமாக அதற்கான குறியீட்டை முன்னதாக துளைக்கவேண்டும்.

இதுவரை நாம் பார்த்தவை யாவும் 5 தாரைக் காகித நாடாவைப் பற்றியே. இந்நாடாவின் அகலம் சாதாரணமாக $\frac{11}{16}''$ ஆகும். இதைப் போலவே 6, 7, 8 தாரைகள் உள்ள நாடாக்களும் உள்ளன. இந்நாடாக்கள் $1''$ அகலம் உள்ளவை. 6, 7, 8 தாரைகள் உள்ள நாடாக்களில், 5 தாரை நாடாவில் உள்ளதைப் போல் ஒரு துளைச்சேர்க்கை இரண்டு உருக்களைக் (character) குறிக்க வேண்டிய அவசியமில்லை. கார்டுகளில் உள்ளதைப்போல் ஒவ்வொரு உருவுக்கும் ஒவ்வொரு தனித்த (Uniqne) துளைச் சேர்க்கை பயன்படுத்தப்படுகிறது.

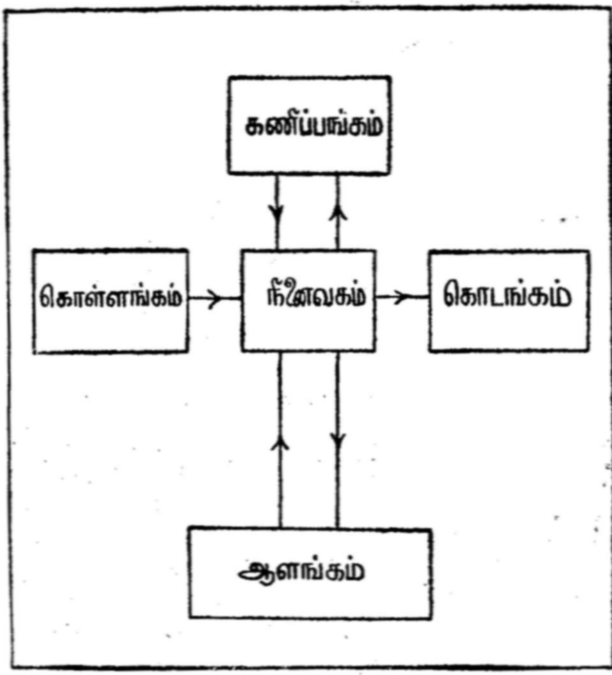
3
கம்ப்யூட்டர் ஒருங்கு
(The Computer System)

இதுவரை நாம் கம்ப்யூட்டர் என்ற சொல்லை மிகவும் பொதுப்படையாகவே பயன்படுத்தி வந்திருக்கிறோம். கம்ப்யூட்டர் என்பது ஒரு தனிப்பட்ட ஒற்றை இயந்திரம் அல்ல; ஒன்றுக்கு மேற்பட்ட மூன்று, நான்கு (சில சமயங்களில் நான்குக்கும் மேற்பட்டு) இயந்திரங்களைக் கொண்ட கூட்டு அமைப்பு ஆகும். இந்த கூட்டமைப்பில் உள்ள இயந்திரங்கள் தனித்தனியாக இருப்பினும் அவை ஒன்றோடொன்று மின் கம்பிகளால் இணைக்கப்பட்டுள்ளன. இத்தகைய ஒரு கூட்டமைப்பை நாம் 'கம்ப்யூட்டர் ஒருங்கு' என்று அழைப்போம்.

கம்ப்யூட்டர் ஒருங்கில் முதன்மையான அங்கங்கள் (i) நினைவகம் Memory (ii) கணிப்பங்கம் Arithmetic unit (iii) ஆளங்கம் (control) (iv) கொள்ளங்கம் (Input unit) (v) கொடங்கம் (output unit) ஆகியவை. இவ்வங்கங்களுககு இடையே உள்ள அடிப்படை இணைப்பு முறையைப் பற்றிக் சிறிது பார்ப்போம்.

கொள்ளங்கம் (Input unit)

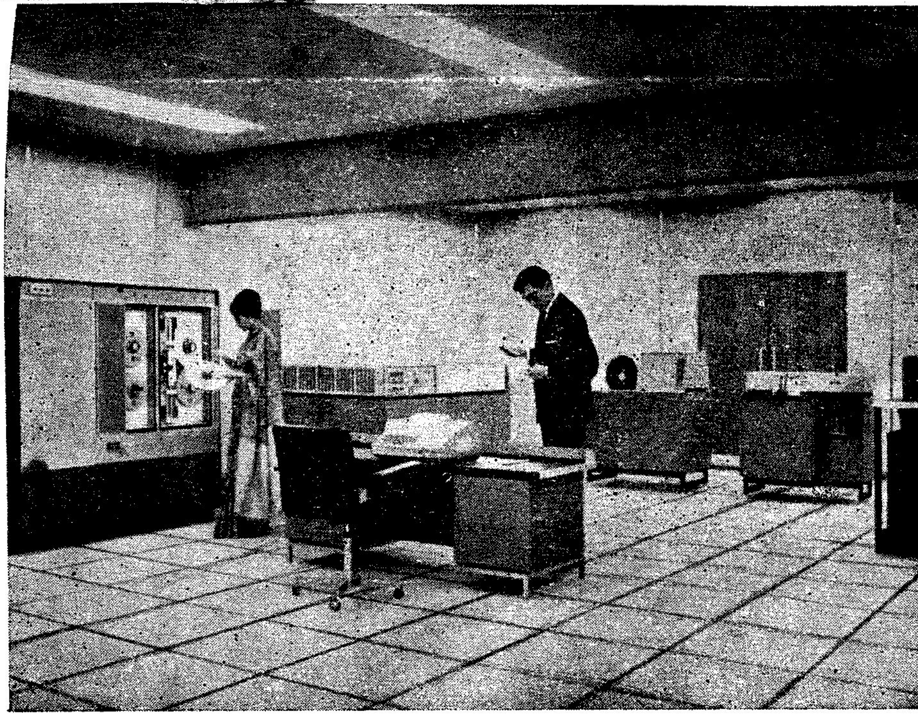
கம்ப்யூட்டர் ஒரு கணிப்பிற்கு தேவையான ஆணைகளையும் அடிப்படை விவரங்களையும் இந்த அங்கத்தின் மூலமே பெறுகிறது. 'கொள்ளங்கம்' பல வகைப்படும். கம்ப்யூட்டருக்கான செய்திகள் கார்டுகளில் துளைக்கப் பட்டிருப்பின் அவற்றைப் படிப்பதற்காகக் கார்டு "படிப்பொறி" (Card Reader) உள்ளது. இக்கார்டு படிப்பொறிகளின் படிப்பு வேகம் நிமிடத்திற்கு 200 முதல் 2000 கார்டுகள். சில படிப்பொறிகள் நிமிடத்திற்கு 200 கார்டுகளே படிக்க வல்லன. மற்றும் சில 1000 கார்டுகளை ஒரு நிமிடத்தில் படிக்கின்றன. நிமிடத்திற்கு 2000 கார்டுகளைப் படிக்க வல்ல படிப் பொறிகளும் உண்டு.



படம் 3.0

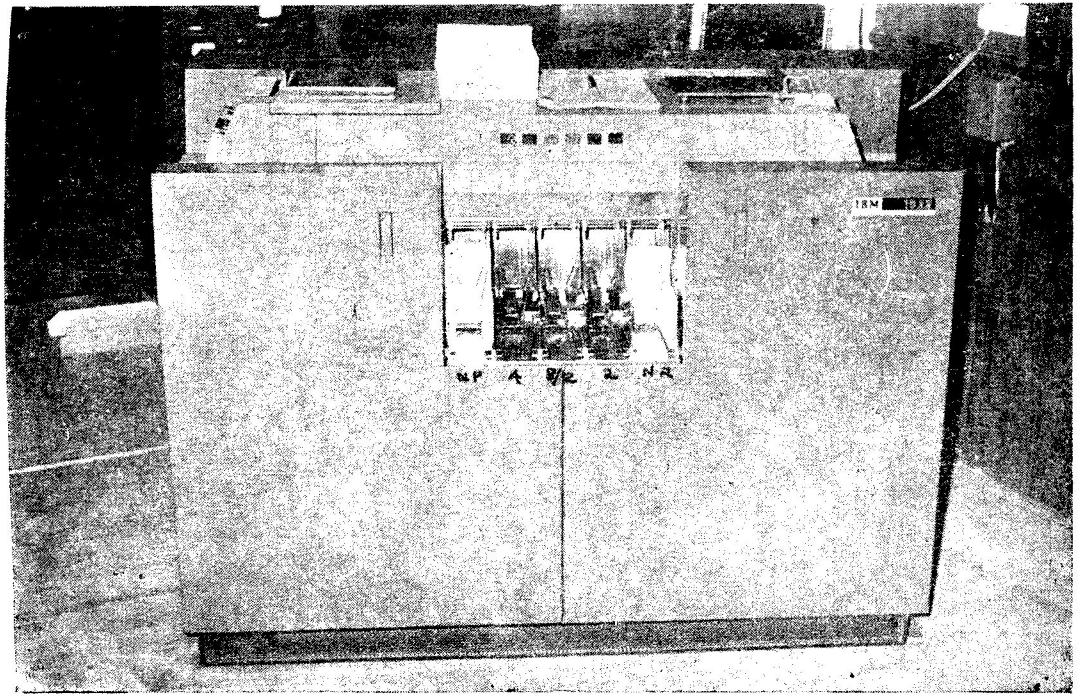
ஒரு கம்ப்யூட்டர் ஒருங்கின் அடிப்படை அங்கங்கள்

படிப் பொறியில் உள்ள நுழை வாயிலில் நமது கார்டுக் கற்றையை (Card Deck) மேல் கார்டு முதலில் உள்ளே போகுமாறு தலைகீழாகத் திருப்பி வைக்கிறோம். படிப்பொறி படிக்க ஆரம்பிக்கும் போது முதல் கார்டை உள்ளிழுக்கிறது. இந்தக் கார்டு உள்ளே சென்றதும் ஒரு பித்தளை உருளைக்கும் மின் முனைகள் (Electric nodes) நிறைந்த ஒரு கட்டைக்கும் இடையே செல்லுகிறது. கார்டு காகிதத்தால் ஆனதனாலும், காகிதத்தினூடே மின்சாரம் செல்லாதாகையாலும் உருளைக்கும், கட்டைக்கும் இடையே கார்டு செல்லும் போது மின் ஓட்டம் தடைப் படுகிறது. ஆனால் துளைகள் எதிர்ப்பட்டதும், இத்துளையினூடே பித்தளை உருளைக்கும் கட்டையிலுள்ள ஒரு மின் முனைக்கும் தொடர்பு ஏற்பட்டு மின்சாரம் பாய்கிறது. கார்டு நகர்ந்து கொண்டேயிருப்பதால் பாயும் மின்சாரம் தொடர்ச்சியாகப் பாயாமல் துடிப்பாகச் (Pulse) செல்லுகிறது. கார்டின் துளையமைப்பிற்கு ஏற்றவாறு இந்தத் துடிப்புகள் கம்ப்யூட்டரின் நினைவகத்தின் மின்காந்த வளையங்களில் செலுத்தப்



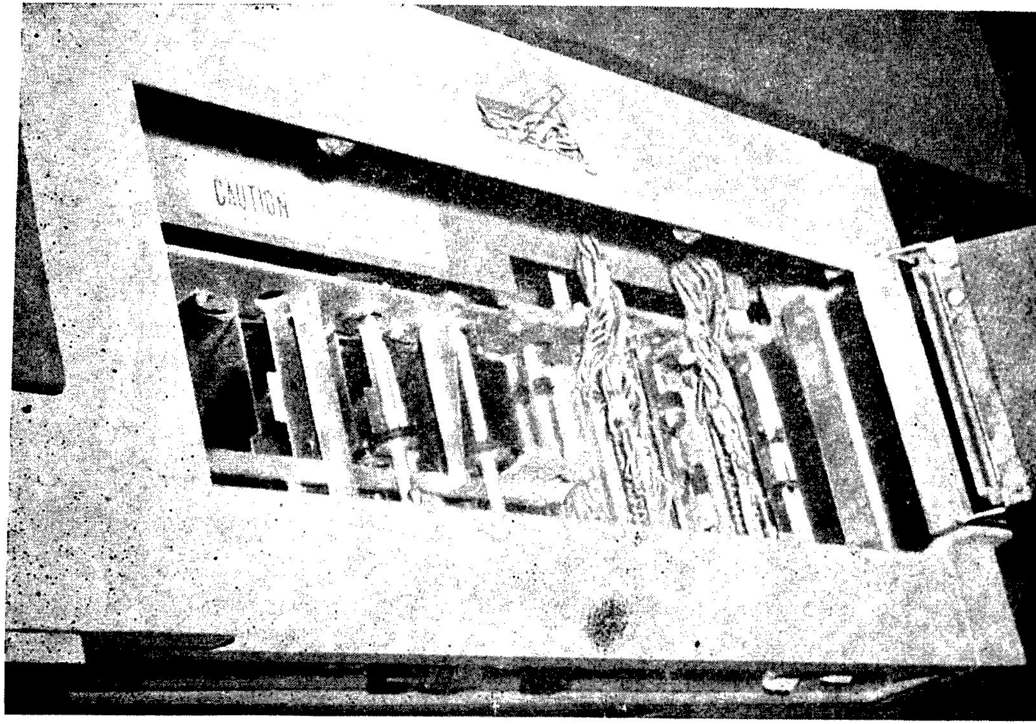
படம் 3.0a
கம்ப்யூட்டர் ஒருங்கு

படத்தில் காட்டப்பட்டுள்ளது ICL 1901-*A* கம்ப்யூட்டர் ஒருங்கு ஆகும். ICL என்பது International Computers Ltd. என்பதன் சுருக்கம். மேல் காணும் கம்ப்யூட்டர் பெங்களூரில் பாரத எலெக்ட்ரானிக்ஸ் நிறுவனம், ICL இவற்றின் கூட்டு முயற்சியால் உற்பத்தி செய்யப்படுகிறது.



படம் 3.1

IBM 1622 கார்டு படிப் பொறி



படம் 3.2

IBM 1622 படிப் பொறியின் படிப்பிடங்கள்

பெற்று, ஏற்ற முறையில் (மின் காந்த வடிவில்) தகுந்த உருக்களை (Characters) நினைவிலிருத்துகின்றன.

சில படிப்பொறிகளில், கார்டு, அகலவாக்கில் படிக்கப் படுகின்றது. இம்முறைப்படி கார்டில் உள்ள 80 நெட்டு வரிசை களும் (columns) ஒருங்கே படிக்கப் பெறுகின்றன. இன்னும் சில படிப் பொறிகளில் கார்டுகள் நீளவாக்கில் நகரும். இம்முறைப் படி (முதல் நெட்டு வரிசை முன் செல்ல) கார்டில் துளையிடப் பட்டுள்ள செய்திகள் ஒவ்வொரு **நெட்டு வரிசையாகப்** (column-wise) படிக்கப் பெறுகின்றன. இது கார்டில் உள்ள உருக்களை (characters) ஒவ்வொன்றாக படிப்பதற்கு ஈடாகும்.

படிப் பொறிகளில் ஒவ்வொரு கார்டும் இரண்டு தனித் தனி இடங்களில் படிக்கப் பெறுகின்றன. இவ்வாறு இரண்டு இடங்களிலும் படிக்கப்பட்ட செய்தி ஒப்புநோக்கப்பட்டு அவை இரண்டும் ஒத்திருக்கின்றனவா என்று பரிசீலிக்கப்படுகிறது. அவை இரண்டும் ஒத்திருந்தால் மட்டுமே அச்செய்தி கம்ப்யூட் டரின் நினைவகத்திற்கு அனுப்பப்படுகிறது. அவை இரண்டும் மாறுபட்டிருப்பின், படிப் பொறி தனது இயக்கத்தை நிறுத்திக் கொண்டு, படிப்பில் ஏற்பட்ட தடையை நமக்கு அறிவிக்கிறது.

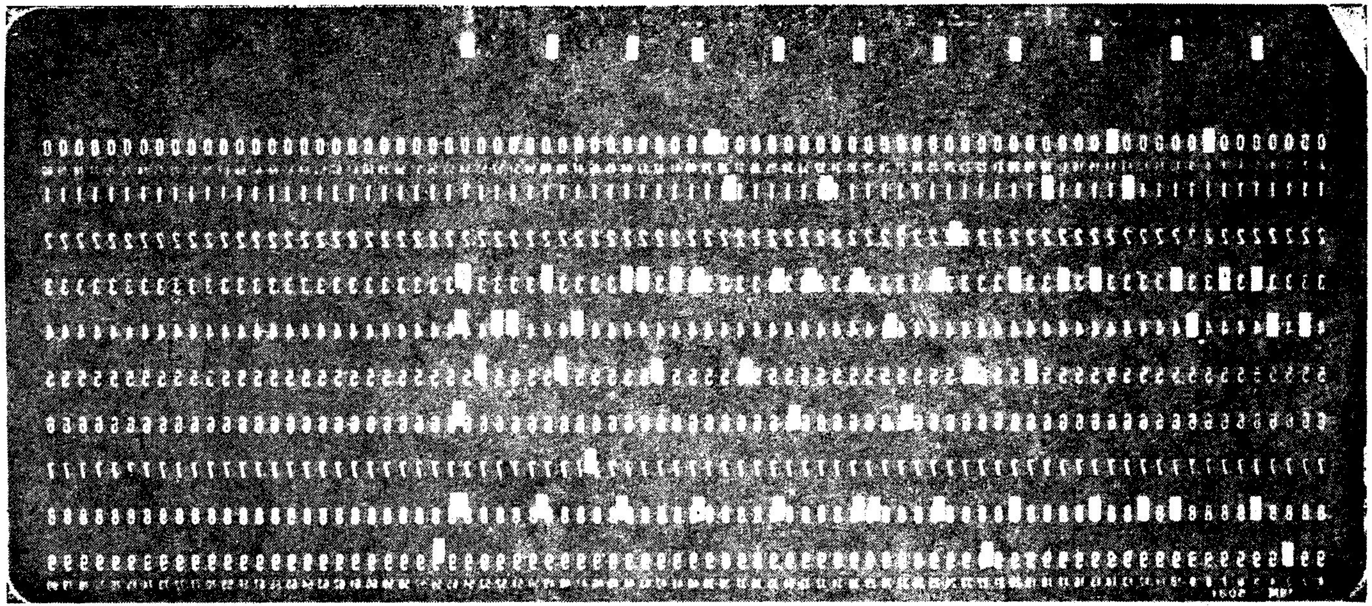
இத்தகைய தடை பல காரணங்களால் ஏற்படலாம். முதலில் படிப் பொறியில் எதற்காக ஒவ்வொரு கார்டும் இரண்டு இடங்களில் படிக்கப் பெறுகிறது என்று பார்ப்போம். படிப் பொறியில் ஒரே ஒரு படிப்பிடம் மட்டும் இருப்பின் சில சமயங் களில் கார்டில் உள்ள துளைகளைப் படிப் பொறி தவருகப் படிக்க நேரலாம். ஏதாவது ஒரு மின் முனையுடன் (Electrical node) தொடர்பு கொண்ட மின்கம்பி தளர்ச்சியற்றிருப்பின் அதன் வழியாகப் போதிய மின்சாரம் பாயாமல் போகலாம். அப்போது அம்முனைக்கு அடியில் கார்டில் துளையிருப்பினும் அதன் வழியாக மின்சாரம் பாயாமல் அத்துளை முற்றும் புறக்கணிக்கப் பட்டு விடும். இதன் காரணமாக படிக்கப் படும் உரு, வேருெரு உருவாக மாறிப்போக ஏதுவுண்டு. எடுத்துக் காட்டாக கார்டில் முதலிடத்தில் (First column) 'S' என்ற எழுத்து துளை வடிவில் இருப்பதாகக் கொள்ளுவோம். கார்டில் 'S' என்ற உருவைக் குறிக்க 'O'விலும் '2'விலும் துளையிடப்பட்டிருக்கும். இப்போது 'O'வின் தொடர்புடைய முனையை இணைக்கும் மின்கம்பி தளர்ச்சி யுற்று இருப்பதாகக் கொள்ளுவோம்; அதன் வழியாகய போதிய மின்சாரம் பாயாமல் 'O'வில் உள்ள துளை புறக்கணிக்கப்படும். அதே சமயத்தில் '2' இடத்தில் உள்ள துளை வழியாக மின்

சாரம் பாயும். அதனுல் 'S' என்ற எழுத்திற்கு மாருக '2' என்ற எண் படிக்கப் பெறும். (கார்டில் '2'ன் மீது இடப்படும் துளை மட்டுமே 2 என்ற எண்ணுக்கான துளைக் குறி (Punch code) என்பதை நினைவு கொள்க.)

மேற்கூறிய தவறு ஏற்படாமல் காக்கவே ஒவ்வொரு கார்டும் இரண்டு இடங்களில் தனித் தனியே படிக்கப் பெற்று, படிக்கப் பட்ட செய்திகள் ஒப்பு நோக்கப் படுகின்றன. ஏதாவது ஒரு படிப்பிடத்தில் (Reading station) ஏதாவது ஒரு மின் கம்பி தளர்வுற்று, படிப்பதில் தவறு ஏற்படின் இப் படிப்பிடத்திலும், மற்ற படிப்பிடங்களிலும் படிக்கப் பட்டவை ஒத்திருக்கா. உடனே படிப் பொறி தாஞகவே நின்று தவறு ஏற்பட்டுள்ளதை நமக்கு அறிவிக்கும்.

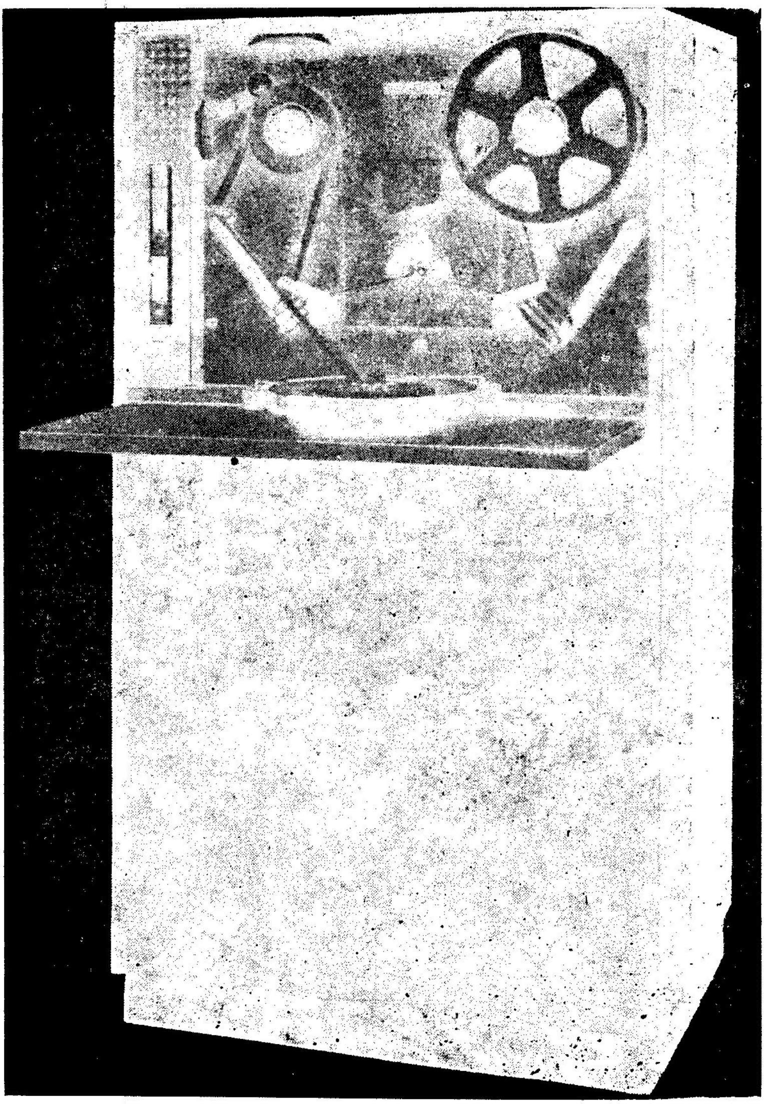
சில சமயங்களில் கார்டில் தவருன (அனுமதிக்க படாத) துளைச் சேர்க்கைகளும் ஏற்பட்டிருக்கலாம். இத்தவறு, துளைப் பொறியில் (punching machine) ஏற்பட்ட தவருலோ அல்லது ஏற்கனவே, துளையிடப் பெற்ற இடங்களின் மீதே மீண்டும் துளை யிடுவதாலோ ஏற்படலாம். எடுத்துக் காட்டாக ஒரு கார்டில் 10 நெட்டுவரிசையில் "8" என்ற எண் ஏற்கனவே துளைக்கப் பட்டிருப்பதாக்கிக் கொள்வோம். அதே இடத்தில் நாம் 9 என்ற எண்ணைக் குறித்தோமாஞல் அவ்விடமும் துளையிடப்படும். ஆக இப்போது கார்டின் 10-வது நெட்டுவரிசையில் (10th column) 8, 9, ஆகிய இரண்டு இடங்களிலும் துளையிடப்பட்டிருக்கும். 8, 9, என்ற இரு துளைச் சேர்க்கைகளைக் கொண்ட உரு (character) ஒன்றும் கிடையாதாகையால், இந்தக் கார்டு படிப் பொறியில் உள்ள படிப் பிடத்தில் செல்லும் போது, இது தவருன துளைச் சேர்க்கை என்று கண்டு கொள்ளப்பட்டு அந்த கார்டு நிராகரிக்கப் படுகிறது.

இன்னும் சில சமயங்களில் கார்டுகளில் துளையிடும் போது, துளை சரியாகக் குறிப்பிட்ட இடத்தில் விழாமல் போகலாம். சற்று மேலோ, கீழோ, இடவலமோ தள்ளி (half punches) துளை விழுந் திருக்கலாம் (இது பெரும்பாலும் துளைப் பொறிகளில் ஏற்படக் கூடிய பழுதுகளால் ஏற்படும்) இத்தகைய துளைகளையுடைய கார்டுகளையும் படிப் பொறி நிராகரித்து விடும். இந்த கார்டுகளை மீண்டும் சரியாக துளையிட்டுத் திரும்பவும் படிப் பொறியில் இடலாம். இத்தகைய தவறுகள் நேரும் போது, தவறுகள் கொண்ட கார்டுகளைக் கவனித்துப் புதிய கார்டுகளைக் கொண்டு சரியாகத் துளையிட்டு திரும்பவும் அந்தக் கார்டு படிப் பொறியைப்

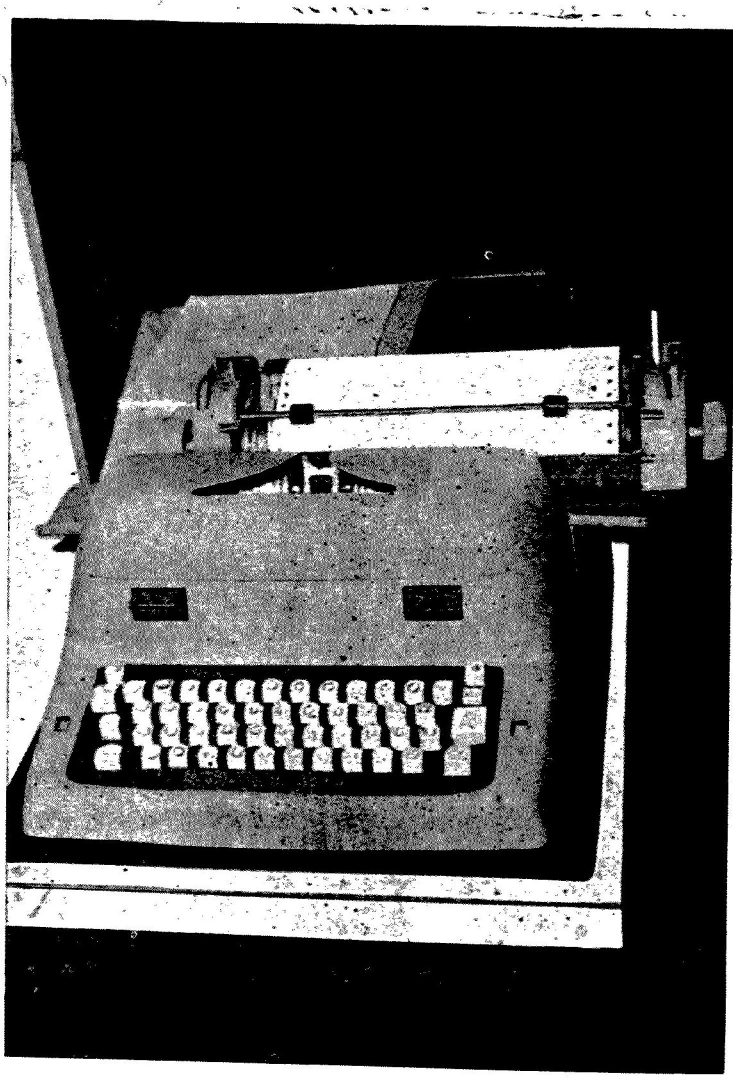


படம் 3.3

சரியான முறையில் துளைக்கப்படாத கார்டு



படம் 3.4
காகித நாடாப் படிப் பொறி



படம் 3.5

IBM 1620-ன் அச்செழுத்துப் பொறி

(Console Typewriter - IBM 1620)

போலவே, மற்றொரு முக்கியமான கொள்ளங்கம், காகித நாடா வாகும். காகித நாடாவில் எவ்வாறு, உருக்கள் துளைச் சேர்க்கை களால் குறிக்கப் படுகின்றன என்பதை முன்னரே பார்த்தோம். அடிப்படை அமைப்பில் கார்டு—படிப் பொறிக்கும் (Card Reader) காகித நாடா— படிப் பொறிக்கும் பெரிய வேறுபாடு ஒன்றும் இல்லை. கார்டு படிப் பொறியில் எவ்வாறு துளைகள் வழியாக மின் துடிப்புகள் செலுத்தப்பட்டு, கம்ப்யூட்டரின் நினைவகத்தில் மின் காந்த வடிவில் உருக்கள் சேர்க்க படுகின்றனவோ அதே முறையில் தான் காகித நாடாக்களைப் படிக்கும் பொறிகளும் இயங்குகின்றன.

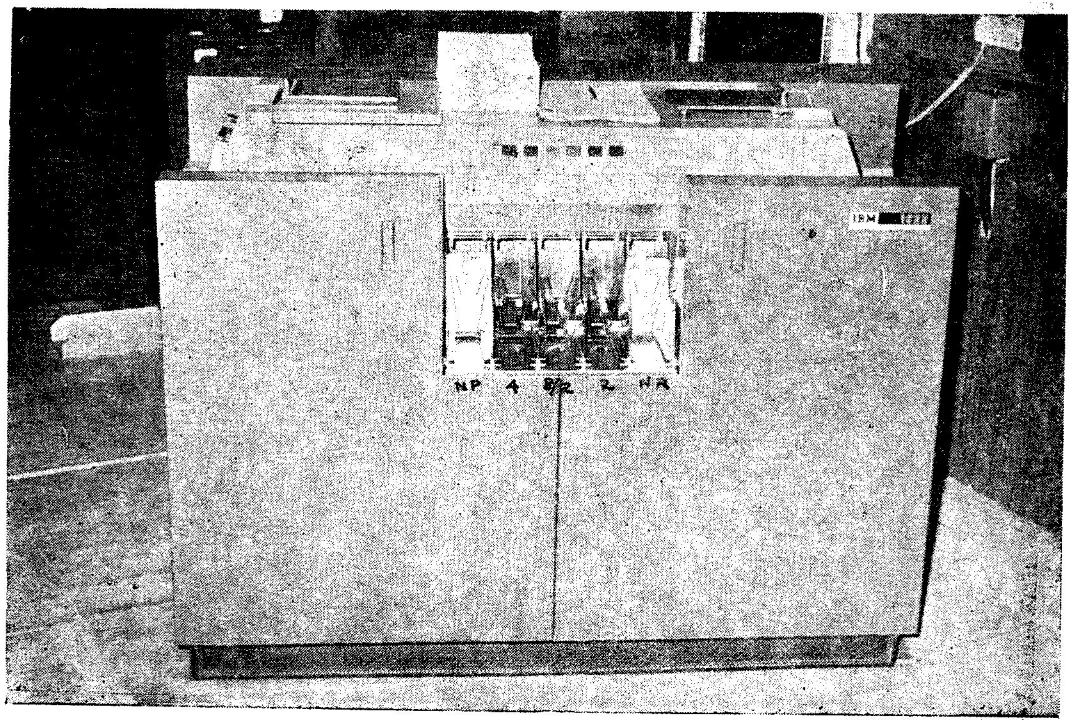
பெரும்பாலான கம்ப்யூட்டர்களில் தட்டெழுத்துப் பொறி (Type Writer,) ஒன்றும் இணைக்கப்பட்டிருக்கும். சாதாரணத் தட்டெழுத்துப் பொறியை இயக்குவதைப் போலவே இதையும் இயக்கலாம். இப்பொறியில் கம்ப்யூட்டருக்கான ஆணைகளையும் அடிப்படை விவரங்களையும் நாம் தட்டும் போது அவ்வாணைகளும் விவரங்களும் தட்டெழுத்துப் பொறியில் செருகப்பட்டுள்ள காகிதத்தில் பொறிக்கப்படுகின்றன. அதுவுமின்றி இவ்வெழுத்துக்கள், பைனரிக் குறியாக மாற்றப்பட்டு கம்ப்யூட்டரின் நினைவகத்தில் சேர்க்கப் படுகிறது. இத்தட்டெழுத்துப் பொறியையும் ஒரு கொள்ளங்கம் எனக் கொள்ளலாம். ஆனால் பெரும்பாலும் இப் பொறியை ஒரு கொள்ளங்கமாகப் பயன் படுத்துவதில்லை. இதற்குக் காரணம் இப்பொறி மூலம் நாம் கம்ப்யூட்டருக்கு விவரங்களைக் கொடுக்க ஆரம்பித்தால் அதிகப்படியான கால தாமதம் ஆவதே.

துளையிட்ட கார்டுகளைப் படிக்கும் கார்டு படிப் பொறி ஒரு நிமிடத்திற்கு 200 கார்டுகள் படிப்பதாயினும் அது நொடிக்கு $200 \times 80 = 1600$ எழுத்துக்கள் ஆகிறது. இந்த வேகத்தில் கம்ப்யூட்டரின் தட்டெழுத்துப் பொறியை இயக்க வல்லார் யார்? அதுவுமின்றி தட்டெழுத்துப் பொறி, நமது இயக்கத்தில் (Manual Mode) இருக்கும் போது கம்ப்யூட்டர் சும்மா இருக்க வேண்டியுள்ளது. இது வீண் கால விரைய மாகும். கம்ப்யூட்டர் விலை மிகுந்த சாதனமாதலின் அது இயங்காமல் விரய மாகும். ஒவ்வொரு நொடியும் குறிப்பிடத்தக்க பொருள் நட்டத்தை விளைவிக்கும். இக் காரணத்தால் கம்ப்யூட்டரின் தட்டெழுத்துப் பொறியைப் பயன்படுத்தி, யாரும் கட்டளைகளையோ அல்லது பெருமளவில் அடிப்படை விவரங்களையோ கொடுப்பதில்லை.

இத் தட்டெழுத்துப் பொறி, கம்ப்யூட்டராலும் பயன்படுத்தப் படுகிறது. ஒரு புரோகிராமின் ஆய்விநூடே நமக்குக் கிடைக்க வேண்டிய சிறு சிறு இடை முடிவுகளை (Intermediate Results) இப் பொறி மூலம் பொறிக்குமாறு நமது புரோகிராமில் ஆணை களைச் சேர்க்கலாம். அதுவுமல்லாமல் நமது புரோகிராம்களை மொழி பெயர்த்துக் கொண்டிருக்கும் போது (Compiling) (பொறி மொழியில்) நமது புரோகிராம்களில் இலக்கணப் பிழைகள் இருப்பின் அவற்றை நமக்கு அறிவிக்கவும் இப் பொறி பயன் படுகிறது.

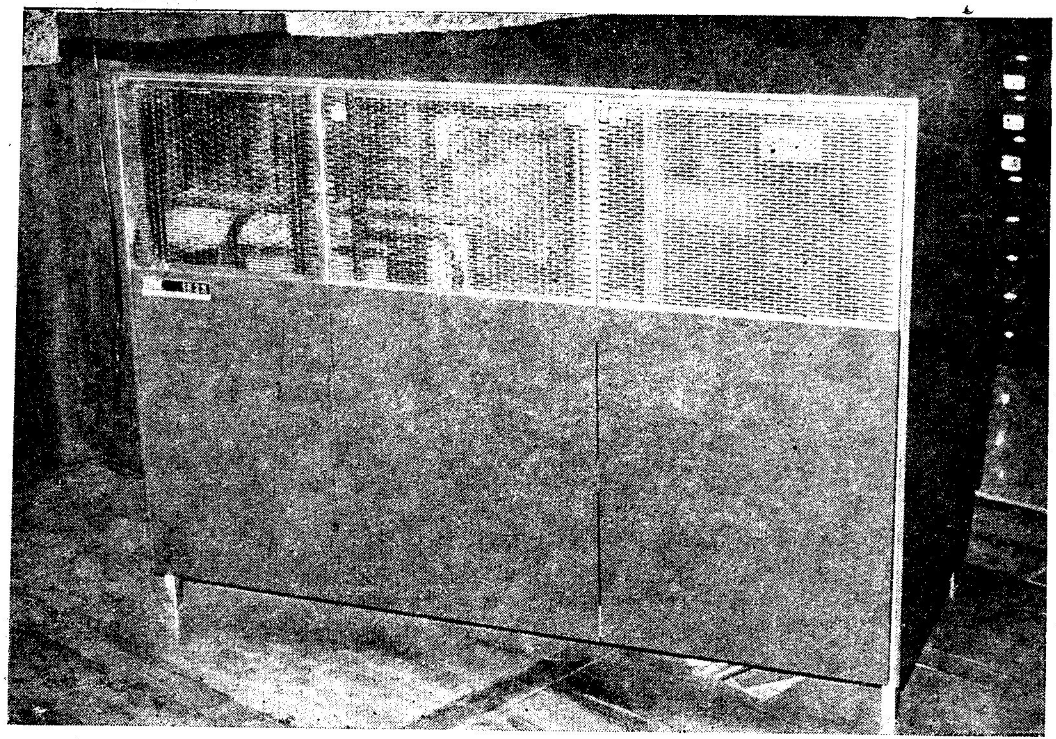
கொடங்கம் :— (Output Unit)

கம்ப்யூட்டர் தனது கணிப்பிற்கான அடிப்படை விவரங் களையும், கட்டளைகளையும் நம்மிடமிருந்து கொள்ளுவதற்காகப் பயன்படும் அங்கத்தை கொள்ளங்கம் என்று கூறினோம். அதுவே போல் கணிப்பு முடிந்த பிறகு இறுதி விடைகளை கம்ப்யூட்ட ரிடமிருந்து நமக்குக் கொடுக்கும் அங்கங்களும் தேவைப்படு கின்றன. இவற்றைக் கொடங்கம் (Output unit) என்று அழைப்போம். எவ்வாறு கம்ப்யூட்டருக்கான செய்திகளைக் கார்டுகளில் துளைக் குறிகள் ஏற்படுத்தி அவற்றின் மூலம் அளிக் கிரேமோ அதே போல் கம்ப்யூட்டரும் தான் அளிக்கும் விடை களையும், பிற விவரங்களையும் கார்டுகளில் துளையிட்டு அளிக்கும். இதற்காக துளைப் பொறி (Punching unit) ஒன்றும் கம்ப்யூட்ட ருடன் இணைக்கப் பட்டிருக்கும். இத்துளைப் பொறியில் ஓரிடத் தில் வெற்றுக் கார்டுகள் (துளையிடப் படாத கார்டுகள்) வெளிப் புறமாக வைக்கப் பட்டிருக்கும். கம்ப்யூட்டர் நமக்கு ஏதாவது ஒரு விவரத்தைக் கார்டு மூலம் அளிக்க வேண்டுமாயின் இக் கார்டுகள் துளைக்கருவியின் உள் இழுக்கப் பட்டு, உள்ளே யிருக்கும் துளை அச்சு மூலம் (Punching die) உரிய துளைகளை ஏற்படுத்திப் பின்னர் துளைக்கப் பட்ட அக்கார்டை வெளியே தள்ளி விடும். பெரும்பாலும் கார்டு படிப் பொறியும் கார்டு துளைப் பொறியும் ஒன்று சேர்ந்து ஒரு அங்கமாகவே (கொள், கொடு) அங்கமாக (Input/output unit) இருக்கும். படம் 3.5-ல் வலது புறம் படிப் பொறியும், இடது புறம் துளைப் பொறியும் அமைந்துள்ளன. இவை யிரண்டும் உள்ளடக்கப் பட்ட அமைப்பே படம் 3.6-ல் காட்டப் பட்டுள்ளது. இப்படத்தில் நடுப்பகுதியில் ஐந்து அறைகள் இருப்பதை நோக்குக. இவ் வைந்து அறைகளுக்கும் முறையே இடமிருந்து வலமாக N-P. (Normal Punch), 4, 8/2, 2, N-R. (Normal Read) என்று பெயரிடுவோம்.



படம் 3.6

படி/துளைப் பொறியின் சேர்ப்பறைகள்



படம் 3.7

IBM 1621—உதவி நினைவகம்

படி / துளைப் பொறிகளில் படிக்கும் பொறியில் படிக்கப்பட்ட கார்டுகள் வந்து விழுவதற்கும், அதே போல் துளைப் பொறியில் துளைக்கப்பட்ட கார்டு வந்து விழுவதற்கும் முறையே NR, NP என்ற அறைகள் (Pockets) உள்ளன. மீதமுள்ள மூன்று அறைகளில் குறிப்பிட்ட சில கார்டுகளை மட்டும் இந்த மூன்று அறைகளிலும் தனியாகச் சேர்க்கவும் வசதியுண்டு. படிப் பொறியின் பக்கம் உள்ள இரண்டு அறைகளும் (அறை 1 அறை N. R.) படிப் பொறியால் பயன்படுத்தப்படும் துளைப் பொறியின் பக்கம் இருக்கும் அறைகள் இரண்டும் (அறை 4, அறை NP) துளைப் பொறியால் பயன் படுத்திக் கொள்ளப்படும். நடுவில் உள்ள (மூன்றாவதாக உள்ள அறை) அறை படிப் பொறி, துளைப் பொறி இரண்டுக்கும் சொந்தமானது. எடுத்துக் காட்டாக சில கார்டுகளை படித்த உடன் 80-வது இடத்தில் * குறியுள்ள கார்டுகளை மட்டும் அறை எண் 1-ல் சேர்க்கும்படிக் கம்ப்யூட்டருக்கு கட்டளையிட்டோமானால், கற்றையிலுள்ள-கார்டு ஒவ்வொன்றும் படிக்கப்படும் போது, 80-ம் இடத்தில் * குறி உள்ளதா என்றும் சோதிக்கப்படும். 80-ம் இடத்தில் * குறியிருப்பின் அது அறை எண் 1-க்கு தள்ளப்படும்; இல்லையாயின் அது 'NR' அறையில் சேகரிக்கப்படும். இம்மாதிரியே கார்டுகளைத் துளைக்கும் துளைப் பொறியில் இருந்து வெளி வரும் கார்டுகளையும் உரிய கட்டளைகள் மூலம் தனித்தனி அறைகளில் NP அல்லது 4 அல்லது 8/2 என்ற அறைகளில் சேகரிக்க இயலும்.

நினைவகம் :—

நினைவகத்தைப் பற்றி ஏற்கனவே நாம் விரிவாகப் பார்த்துள்ளோம். நினைவகத்திற்கு வெளி உலகிலிருந்து வர வேண்டிய செய்திகள் யாவும் ஏதாவது ஒரு கொள்ளங்கத்தின் வழியாக நினைவகத்தில் வந்து சேர்க்கப்படுகின்றன. கணிப்பினிடையே ஏற்படும் மற்ற இடைநிலை முடிவுகள் (Intermediate results) நினைவகத்தில் தான் இருத்திவைக்கப்படுகின்றன. கணிப்பு முடிந்த பின்னரும் நமக்கு கிடைக்க வேண்டிய விடைகளும் இன்ன பிற செய்திகளும் இங்கிருந்தே தொடங்கங்களுக்குச் செல்லுகின்றன. சில சமயங்களில் குறிப்பிட்ட பெரிய கணிப்புகளில், கம்ப்யூட்டரின் நினைவகம் போதாமல் போகலாம். அச்சமயங்களில் பயன் படுத்துவதற்கென்று, அதிகப்படியான நினைவகத்தையும் ஒரு கம்ப்யூட்டர் அமைப்பில் சேர்க்க முடியும். எடுத்துக் காட்டாக IBM 1620 கம்ப்யூட்டரில் 20,000 (20-K என்று அழைப்பர்) நினைவறைகளைக் கொண்ட நினைவகம் உள்ளது. இத்துடன் மேலும் ஒரு 20,000 நினைவறைகளைக்

கொண்ட ஒரு உதவி நினைவகத்தைச் (Auxiliary Memory) சேர்த்து கம்ப்யூட்டரின் நினைவகத்தின் அளவை 40,000 (40-K) ஆக்க முடியும். இத்தகைய வசதி இன்று எல்லா கம்ப்யூட்டர்களிலுமே உள்ளது.

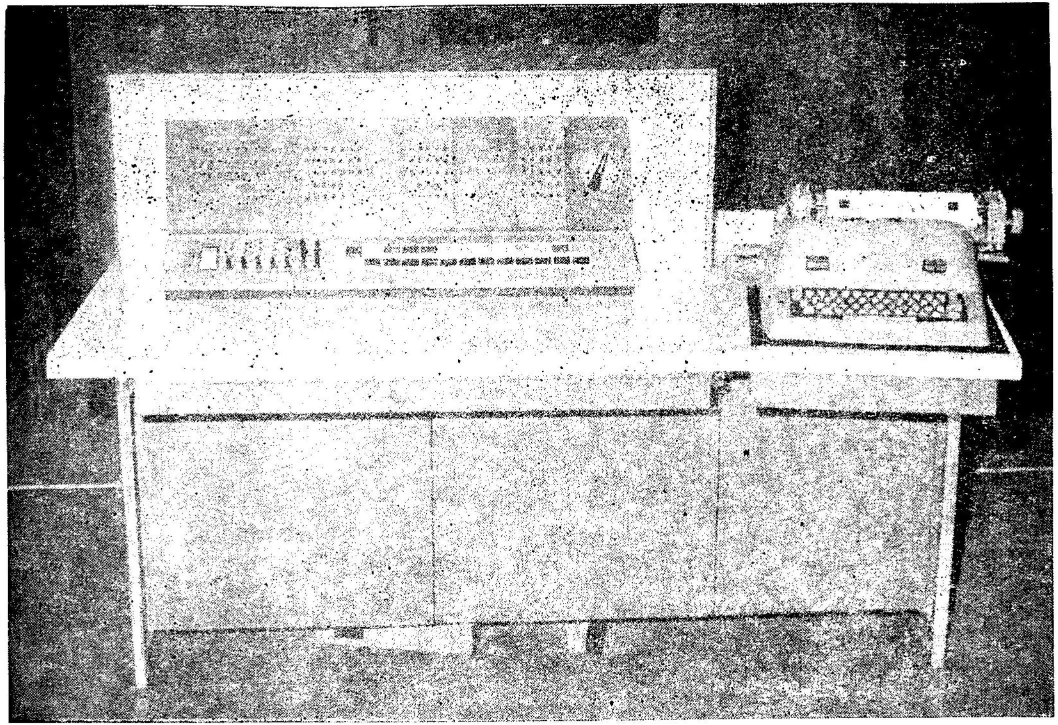
கணிப்பங்கம் (Arithmetic/Logic unit)

இவ்வங்கத்தின் செயலை அதன் பெயரே எடுத்துக் காட்டு கிறது. எந்த ஒரு கணிப்பும்— கூட்டல், கழித்தல், போன்றவை —இவ்வங்கத்தில் தான் நடைபெறும். எடுத்துக் காட்டாக 1738 என்ற எண்ணையும் 17 என்ற எண்ணையும் கூட்ட வேண்டும் என்று வைத்துக் கொள்ளுவோம். (இந்த இரண்டு எண்களும் ஏற்கனவே நினைவகத்தில் இருந்தாக வேண்டும்.) நினைவகத் திலிருந்து இந்த இரண்டு எண்களும் ஒன்றன் பின் ஒன்ருக கணிப் பங்கத்திற்கு எடுத்துச் செல்லப் படுகின்றன. பின்னர் அவ்வங்கத்தில் அவையிரண்டும் கூட்டப் பெற்று, விடையான 1755 என்ற புதிய எண் கணிப்பகத்திலேயே பெறப்படுகிறது. இப்புதிய எண் கணிப்பங்கத்தில் இருந்து நினைவகத்திற்கு எடுத் துச்செல்லப்பட்டு அங்கே இருத்தி வைக்கப்படுகிறது. (Stored) இவ்வாறு கணிப்பங்கம் கம்ப்யூட்டரின் கணிப்பிடமாகப் பயன் படுத்தப் படுகிறது.

ஆளங்கம் (control unit)

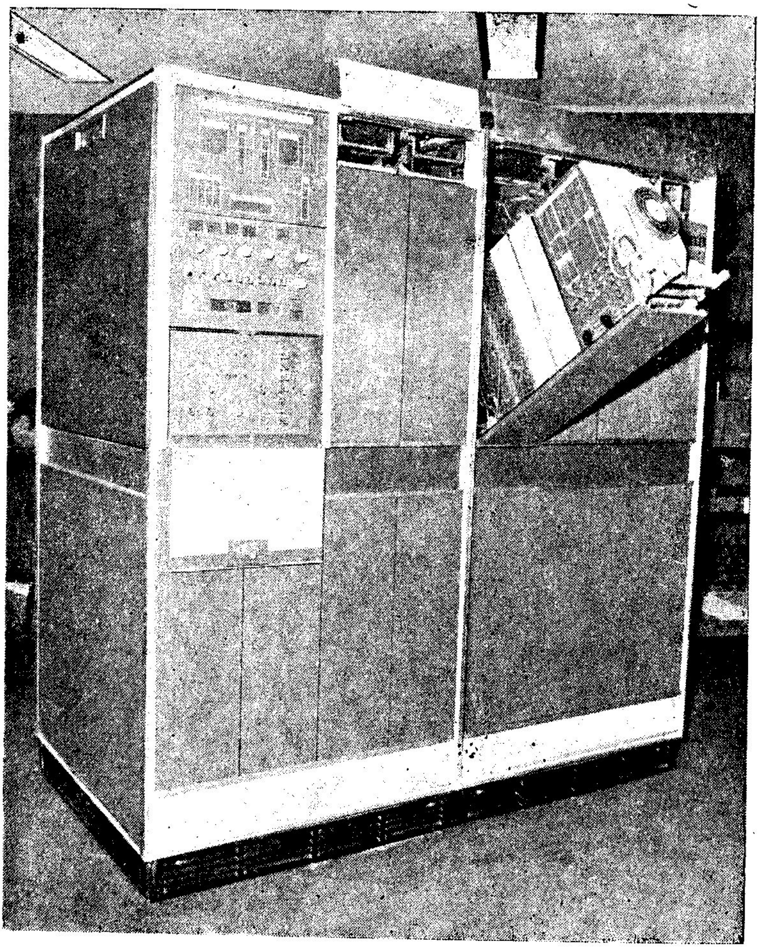
கம்ப்யூட்டர் ஒருங்கில் உள்ள பல்வேறு அங்கங்களின் இயக்கங்களையும் கட்டுப்படுத்தி முறைப்படி இயங்கச் செய்து ஆள்வது இவ்வங்கமே. எடுத்துக் காட்டாக ஒரு துளைக் கார்டைப் படித்து அதிலுள்ள செய்தியை நினைவகத்தில் சேர்க்கும் பணியைப் பார்ப்போம். முதலில் துளைக் கார்டை படிக்கும்படி படிப்பொறியை ஆளங்கம் தூண்டுகிறது. அதே சமயத்தில் படிப் பொறியிலிருந்து வரப் போகும் செய்தியை ஏற்றுக் கொள்ளும் படி நினைவகத்தையும் தூண்டுகிறது. இந்த இரண்டு பொறி களும் ஒன்று பட்டு செயல் படுவதன் மூலம் கார்டு படிக்கப்பட்டு அதிலுள்ள செய்தி நினைவகத்தில் சேர்க்கப்படுகிறது. கார்டு படிக்கப் பெறும் போது அதை ஏற்றுக் கொள்ள நினைவகம் தயா ராக இல்லாமல் இருந்தாலோ, அல்லது நினைவகம் தயாராக இருக்கும் போது, படிப்பு நிகழாமல் இருந்தாலோ, மேற் கூறிய பணி நிறைவேருது.

அதுபோல கணிப்புச் செய்ய வேண்டிய நிலையில் நினை வகத்தையும் கணிப் பங்கத்தையும் ஒன்றிச் செயல்படுமாறு



படம் 3.8

IBM *1401*-ன் தலைமை ஆய்வுப் பொறி (IBM 1401 Central Processing Unit—CPU)



படம் 3.8a

IBM *1620* கம்ப்யூட்டரின் தலைமை ஆய்வுப் பொறி
(ஆய்வுப் பொறியுடன் இணைக்கப்பட்டிருக்கும் அச்சுப்
பொறியை நோக்குக)

செய்வதும் ஆளங்கமே. எடுத்துக் காட்டாக நினைவகத்திலுள்ள 171 என்ற எண்ணையும் 231 என்ற எண்ணையும் கூட்டவேண்டுமாயின் ஆளங்கம் நினைவகத்தைச் செயல்படுத்தி இவ்வெண்களை கணிப்பங்கத்திற்கு அனுப்பச் செய்கிறது. அதே சமயத்தில் கணிப்பங்கத்தையும் இவ்விரண்டு எண்களையும் ஏற்குமாறு செயல்படுத்துகிறது. பின்னர் கணிப்பங்கத்தில் உள்ள கூட்டலைச் செய்வதற்கான மின் வழிகளைத் (Circuits) தூண்டி இவ்விரண்டு எண்களையும் கூட்டுமாறு செய்கிறது. கூட்டல் முடிந்தவுடன், கணிப்பங்கத்தில் உள்ள விடையை நினைவகத்திற்கு அனுப்பி விடுமாறு கணிப்பங்கத்தையும்; அங்கிருந்து வரும் எண்ணை ஏற்று சேர்த்துக் கொள்ளுமாறு நினைவகத்தையும் ஆளங்கம் தூண்டுகிறது.

இவ்வாறு கூட்டி வந்த விடையை நமக்கு அச்சிட்டுத் தெரிவிக்க வேண்டியிருப்பதாக வைத்துக் கொள்ளுவோம். உடனே நினைவகத்தைத் தூண்டி இவ்வெண்ணை உரிய கொடங்கத்திற்கு (அச்சுப் பொறிக்கு) அனுப்புமாறு ஆளங்கம் தூண்டுகிறது. அதே சமயத்தில் அச்சுப் பொறியும் இயக்கப்பட்டு அது நினைவகத்திலிருந்து வரும் எண்ணை ஏற்று, அச்சிட்டு நமக்கு அளிக்கிறது. இவ்வாறாகக் கம்ப்யூட்டர் ஒருங்கில் உள்ள ஒவ்வொரு அங்கமும் உரிய நேரத்தில் ஒன்றுபட்டு இயங்கிச் செயல் பட ஆளங்கம் காரணமாகிறது.

சாதாரணமாக, நினைவகம், கணிப்பங்கம், ஆளங்கம் ஆகிய மூன்றும் ஒன்று சேர்ந்து ஒரே அமைப்பாக இருக்கும். இவ்வமைப்பு சென்ட்ரல் ப்ராஸஸிங் யூனிட் (central processing unit) அல்லது தலைமை ஆய்வுப்பொறி எனப்படும். மனித உடலமைப்பில் மூளைக்கு உள்ள முக்கியத்துவம் ஒரு கம்ப்யூட்டர் ஒருங்கில் சென்ட்ரல் ப்ராஸஸிங் யூனிட்டிற்கு உண்டு. கம்ப்யூட்டர் ஒருங்கில் மற்ற அங்கங்கள் மனிதரின் கண், மூக்கு, செவி, கை, கால்களைப் போன்றவை.

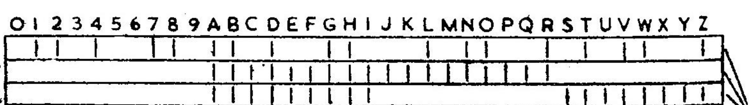
வேறு சில கொள்/கொடு துறைகளும் அங்கங்களும்

கார்டு, காகித நாடா ஆகியவற்றைக் கொண்டு கம்ப்யூட்டருக்கான விவரங்களை அளிக்கவும், கம்ப்யூட்டரிடமிருந்து நமக்குத் தேவையான முடிவுகளையும், விவரங்களையும் பெறவும் முடியும் என்று பார்த்தோம். ஆனால் இவற்றின் வழியாகச் செய்திகளைக் கொடுக்கவோ கொள்ளவோ ஆகும் நேரம், கம்ப்யூட்டரின் கணிப்பு வேகத்தை ஒப்பு நோக்க, மிகவும் அதிகமாகவும், மிக வேகமாக இயங்கவல்ல கணிப்பங்கம் கொண்ட கம்ப்யூட்டர் அமைப்பில் மெதுவாக இயங்கும் கொள்/கொடு

அங்கங்களைச் சேர்ப்பின் கணிப்பு வேகமே பயனற்றுப்போகிறது. இக்காரணத்தால் வேகமாக இயங்க வல்ல கொள்/கொடு துறைகளும் (Media) சாதனங்களும் தேவைப்படுகின்றன. இத்தகைய சாதனங்கள் சிலவற்றை இப்பகுதியில் பார்ப்போம்.

காந்த நாடா. (Magnetic Tape)

காந்த நாடா, உருவமைப்பில் நாம் சாதாரணமாகப் பயன்படுத்தும், ''டேப் ரெகார்டர்'' நாடாவைப் போலவே பெரும்பாலும் இருக்கும். இவை இரண்டுக்கு மிடையே உள்ள முக்கியமான வேறுபாடுகளாவன : (i) கம்ப்யூட்டர் பயன்படுத்தும் காந்த நாடா 0.5" லிருந்து 3.0" அங்குலம் வரை அகலம் உள்ளது. ஆனால் டேப் ரிகார்டர் நாடா ½" முதல் 1" வரையே அகலம் உள்ளது. (ii) கம்ப்யூட்டர் பயன்படுத்தும் காந்த நாடா சுருணை (Reel) ஒவ்வொன்றிலும் 1,200" முதல் 3000" வரை நாடா இருக்கக்கூடும். பெரும்பாலான சுருணைகளில் 2,400 அடி நாடா காணப்படும். (iii) கம்ப்யூட்டர் நாடா தரத்தில், டேப் ரிகார்டர் நாடாவைவிட மிக மிக உயர்ந்தது. கம்ப்யூட்டரில் பயன்படுத்தும் காந்த நாடா 0.02" பருமன் (Thickness) கொண்டது. ஒரு புறம் பிளாஸ்டிக் படர்ந்தும் மற்றொருபுறம் இரும்பு ஆக்ஸைடு துகள்கள் நிறைந்தும் காணப்படும்.



படம் 3.9

7-தாரை காந்த நாடாவின் குறியீட்டு அமைப்பு

1. தாரைகள் 2. காந்தப் புள்ளிகள்

[நாமனின்மேல் காணப்படும் 0 1 2 3...Y Z ஆகிய உருக்கள் ஒவ்வொன்றும் அதன் அதன் நேர் கீழே காணும் காந்தப் புள்ளிகளின் தொகுதியால் குறிக்கப்படுகின்றன.]

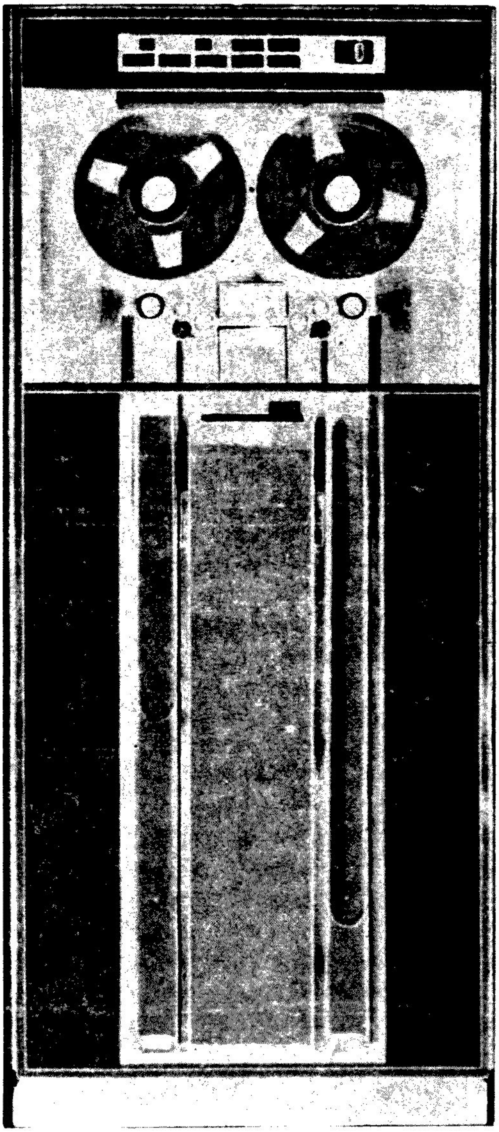
இந்த இரும்பு ஆக்ஸைடு துகள்களே காந்த மேற்க வல்லன. இவற்றில் காந்த மேற்றி, பைனரி முறையில், காகித நாடாவில் உள்ளதைப் போலவே, எண்களும், எழுத்துக்களும் மற்ற குறியீடுகளும், காந்த நாடாவில் ஏற்றப் படுகின்றன. காகித நாடா

வுக்கும், காந்த நாடாவிற்கும் ஓரளவு பைனரியைப் பயன் படுத்தும் முறையில் ஒற்றுமை உண்டு. காகித நாடாவைப் போலவே காந்த நாடாவிலும், ஒவ்வொரு உருவும் (character) நாடாவின் குறுக்காக (across the tape) பைனரிக் குறிகள் (காகித நாடாவில் துளைகளும் காந்த நாடாவில் காந்த புள்ளிகளும்) மூலம் எழுதப் படுகிறது. மின் காந்த நாடாவில், நாடாவின் நீள வாக்கில் பைனரி இடங்களுக்காகவும், பகுதி இடங்களுக்காகவும் தாரைகள் அமைந்துள்ளன. படம். 3.9-ல் 7 - தாரை காந்த நாடாவின் அமைப்பும் 0 முதல் 9 வரை யான இலக்கங்கள், **A.B.C... Z** ஆகிய எழுத்துக்கள் ஆகியவற்றிற்கான பைனரிக் குறியீடுக்களும காட்டப்பட்டுள்ளன. கம்ப்யூட்டரின் காந்த நாடாவிற்கும், சாதாரண டேப் ரிகார்டர் நாடாவிற்கும் ஒரு வித்தி யாசம் உண்டு. டேப் ரிகார்டர் நாடாவில் ஒலிக் குறிகள் ஒரு தொடர்ந்த நீண்ட சங்கேதமாக (code) இருக்கும். கம்ப்யூட்டர் நாடாவில் உள்ள குறியீடுகள் ஒன்றுக்கொன்று இடைவிட்டு உதிரி யாக அமைந்திருக்கும். இவ்வாறு உதிரியாக அமைந்துள்ள குறிப்பீடுகளின் நெருக்கத்தை ஒட்டி அவைகளின் 'அடர்வு' (Density) தீர்மானிக்கப்பெறும். ஒரு அங்குல நாடாவில் 200 உருக்கள், 556 உருக்கள், 1100 உருக்கள்என மூன்று வகை அடர் வுகளில் எண்களும் எழுத்துக்களும் காந்த நாடாவில் எழுதப்படு கின்றன. இம் மூன்று வகை அடர்வுகளில் முதலிரண்டு வகை மட் டுமே பெரிதும் பயன்படுத்தப்படுபவையாம். அவற்றிற்கு முறையே குறை அடர்வு (Low Density) மிகை அடர்வு, (High Density) என்று பெயர். மூன்றும் வகை அடர்வு (அங்குலத்திற்கு 1100 உருக்கள்) அதிமிகை அடர்வு (Super High Density) எனக் கூறப் படும். இம்மிகை அடர்வு முறைப்படி, ஒரு அங்குல காந்த நாடாவில் 1100/80 = 13¾ கார்டுகளில் (80 இடங்களிலும் துளை யிடப்பட்ட) உள்ள செய்தி முழுவதும் எழுதி முடிக்கலாம். இதனால் மிகச்சிறிய இடத்தில் பெரும் அளவு செய்திகளை சேர்த்து வைக்க இயலுகிறது. மற்றொரு முக்கியமான அம்சம், காந்த நாடாவிலிருந்து செய்திகளைக் கம்ப்யூட்டருக்குக் கொண்டு வரு வதிலோ அல்லது கம்ப்யூட்டரிலிருந்து செய்திகளை காந்த நாடா வின் மீது எழுதுவதிலோ உள்ள விரைவு ஆகும். ஒரு விநாடிக்கு 150" நீள காந்த நாடாவிலிருந்து செய்திகளைக் கொள்ளவோ அல்லது அதன் மீது எழுதவோ இயலும். அப்படியானால், ஒரு விநாடிக்கு 1,65,000 எண், எழுத்துக்கள் வரை இந்த நாடாக் களிலிருந்து செய்திகளைப் படிக்க அல்லது அதன் மீது எழுத முடிகிறது. அதாவது ஒவ்வொரு வரியிலும் 40 எழுத்துக்களும் ஒவ்வொரு பக்கத்திலும் 40 (இத்தகைய) வரிகளும் கொண்ட புத்தகம் ஒன்றின் 6188 பக்கங்களை ஒரு நிமிடத்தில் சுமார் 13

அடி நீளமுள்ள காந்த நாடாவில் எழுதி விடலாம். காந்த நாடாவின் விவரங்களைச் சேகரித்து வைப்பதில் இடமும் மிகச் சிக்கனமாக அமைகிறது. சாதரணமாகப் பயன்படுத்தப்படும் மிகை அடர்வு-முறையிலேயேகூட ஒரு சுருணை காந்த நாடாவில் 1,80,000 துளைக் கார்டுகளில் உள்ள செய்திகளைச் சேகரிக்க இயலும். இதே சுருணையில் அதிமிகை அடர்வு முறையில் 3,60,000 கார்டுகளில் உள்ள செய்திகளைச் சேர்க்க இயலும். இதுமட்டுமல்லாமல் காந்த நாடாக்களைத் தயாரிப்பதில் இன்று மேலும், மேலும் மாற்றங்களும், அபிவிருத்திகளும் நடந்து கொண்டே இருக்கின்றன. இதனால் காந்த நாடாவின் அடர்வு எண் மேலும் மேலும் உயர்ந்துகொண்டே போகிறது. ஒரு சுருணையில் (2,400 அடி நாடா) 640,00,000 உருக்களை ஏற்க வல்ல காந்த நாடாக்களும் உண்டு.

கார்டுகளில் விவரங்களைச் சேர்த்து வைப்பதில் நடைமுறையில் சில அசௌகரியங்கள் உள்ளன. முதலாவதாகக் கார்டு அதிக இடத்தை அடைத்துக் கொள்ளுகிறது. இரண்டாவதாக கார்டு மூலம் கம்ப்யூட்டருக்குச் செய்திகளை அளிப்பதில் அதிகப் படியான காலம் செலவாகிறது. மூன்றாவதாக, கார்டு தட்ப வெட்ப நிலையால் எளிதில் பாதிக்கப்படும். இத்தகைய குறைகள் காந்த நாடாவைப் பயன் படுத்துவதில் இல்லை. ஆனால் காந்த நாடாவை கையாளும் போது மிகவும் கவனமாக இருத்தல் அவசியம். காந்த நாடாவின் காந்தப் புள்ளிகள் மிகவும் சிறிய வையாக இருப்பதால், ஒரு சிறு தூசு படிந்தாலும், அது ஒரு காந்தப் புள்ளியை மறைத்துக் கொள்ள வாய்ப்பு ஏற்படும். அந்த நிலையில் நாடா படிக்கப்பட்டால் தவறான விவரங்களே படிக்கப் பெறும். அதுவுமல்லாமல் காந்த நாடாவை, அதை படிக்கும் கருவியில் இணைக்கும் போது நாடாவில் ஏற்படும் மிகச் சிறு சுருக்கமோ, அல்லது இணைப்பவர் கையிலிருந்து நாடாவில் ஒட்டிக் கொள்ளும் சிறு தூசுகளோ தவறான விவரங்களுக்கு ஏதுவாகும்.

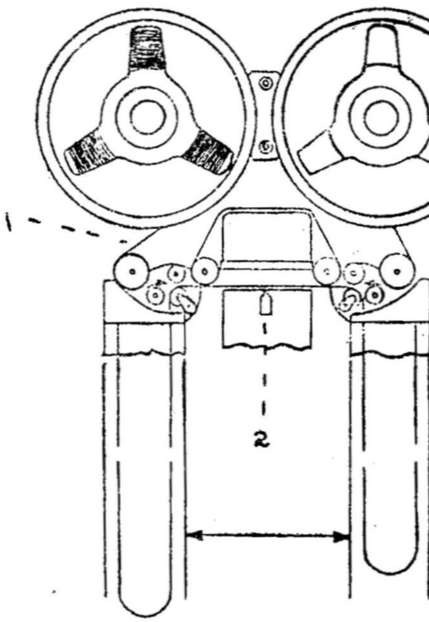
காந்த நாடாவை பயன்படுத்துவதில் ஒன்றை நினைவு வைத்துக் கொள்ள வேண்டும். காந்த நாடாவில் நேரிடையாக விவரங்களை எழுதுவதோ அல்லது அதிலுள்ள விவரங்களைப் படிக்கவோ மனிதர்களால் இயலாது. கம்ப்யூட்டர் மட்டும் தான் காந்த நாடாவைப் படிக்க இயலும்; கம்ப்யூட்டர் தான் அதன் மீது எழுதவும் முடியும். காந்த நாடாவில் ஏற்ற வேண்டிய செய்தி எதுவாயிருப்பினும் அவை முதலில் கார்டுகளில் துளைக்கப் பட்டு, பின்னர் இக்கார்டுகளில் உள்ள செய்தியைக்



படம் 3.10

காந்த நாடாப் பொறி (Magnetic Tape Unit)

கம்ப்யூட்டர் படித்து, நினைவகத்தில் இருத்திக் கொண்டு, அதன் பின்னர், நினைவகத்தில் இருத்திக் கொண்ட இச்செய்தியைக் காந்த நாடாவில் எழுதுகிறது.



படம் 3.11
காந்த நாடா படிப்பொறி

1. காந்த நாடா 2. படி எழுது முகம்

படம்- 3. 11-ல் காந்த நாடாவை இயக்கும் பொறியின் மாதிரி வரைபடம் காட்டப்பட்டுள்ளது.

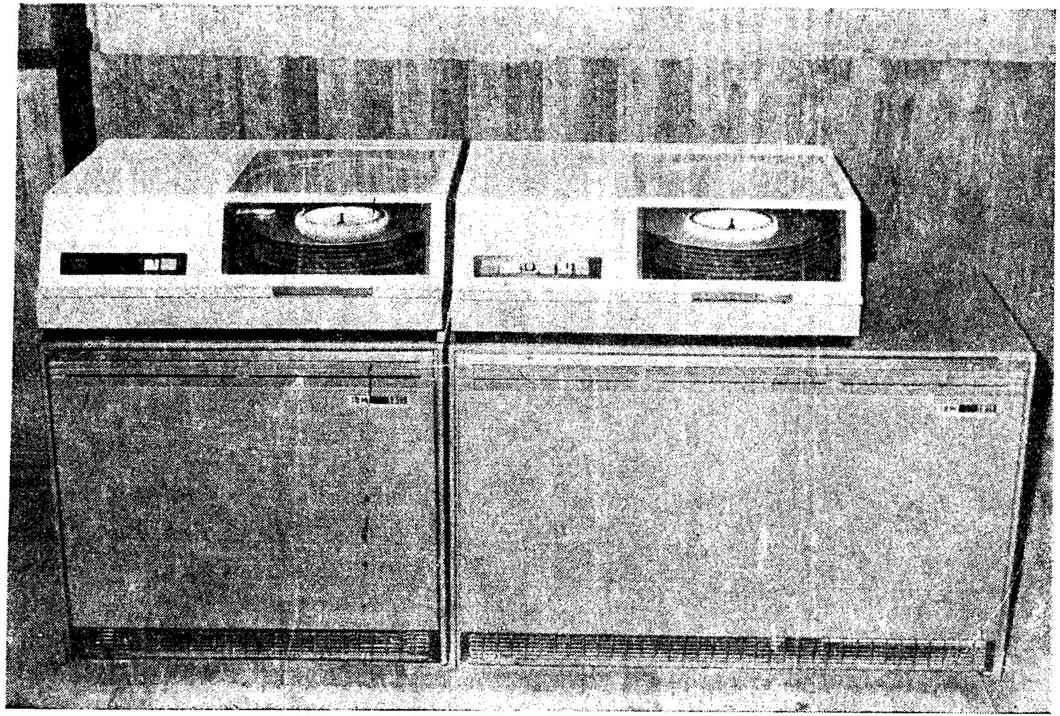
இப்படத்தின் தலைப்புறத்தில் இரண்டு வட்டவடிவமான சுருணைகள் (reels) உள்ளன. அவற்றுள் ஒன்றிலிருந்து (இடப் புறம்) காந்த நாடா கரக்கப்படுகிறது (unreel). மற்றது (வலப் புறம் நாடாவைச் சுற்றிக்கொள்ளுகிறது. இரண்டு சுருணை களுக்கு மிடையே எழுது / படி முனை (Read / Write Head) உள்ளது. இம்முனையில் எழுது முனை (Write Head) ஒன்றும் படி முனை (Read Head) ஒன்றும் உள்ளன. காந்த நாடாவில் கம்ப்யூட்டர் ஒரு செய்தியை எழுத வேண்டுமாயின் இவ்வெழுது

முனை அச்செய்தியை எழுதுகிறது. நாடா கரக்கும் திசையில் எழுது முனைக்கு அடுத்து, படி- முனை உள்ளது. படி- முனை நாடாவில் சற்று முன் எழுதப்பட்டதைப் படித்து, எழுதப்பட வேண்டிய செய்தி சரியான முறையில் எழுதப்பட்டுள்ளதா எனச் சோதிக் கிறது. இச் சோதனை, எழுதும் போது மட்டுமே நடைபெறுகிறது. நாடாவிலிருந்து கம்ப்யூட்டர் ஏதாவது படிக்க வேண்டுமாயின் அப்போது இத்தகைய சோதனை ஒன்றும் நடை பெறுவது கிடையாது.

காந்த நாடாவில் சில அசௌகரியங்களும் உண்டு. எடுத்துக் காட்டாக, ஒரு காந்த நாடா சுருணையில் கடைசிப் பகுதியில் உள்ள விவரம் ஒன்று நமக்கு வேண்டுமாயின், அந்த சுருணை முழுவதும் கரக்கப் (un reel) படவேண்டும். இதனால் கம்ப்யூட்டரின் நேரம் வீணாகும். காந்த நாடாவிலிருந்து விவரங் கள் யாவும் ஒன்றன் பின் ஒன்றாக வரிசையாகத்தான் நாம் பெறவோ அல்லது அதில் எழுதவோ இயலும். இத்தகைய முறைக்கு வரிசை முறை (Sequential Method) என்று பெயர். இந்த முறையில் விவரங்களைச் சேகரித்து வைத்துக் கொள் வதில் சில வகையான கணிப்புக்களில் சிரமம் ஏதும் இருக்காது, எடுத்துக் காட்டாக ஒரு தொழிற் சாலையில் உள்ள 50,000 பேர்களின் மாதச் சம்பள விவரங்களை ஒன்றன் பின் ஒன்றாக காந்த நாடாவில் எழுதிக் கொண்டு கையாள்வதில் சிரமம் ஏது மில்லை. ஏனெனில் ஒவ்வொரு மாதமும் ஒவ்வொரு தொழிலாளி யும் சம்பளம் வாங்கப் போகிறான். அதனால் முதலிலிருந்து ஆரம் பித்து ஒவ்வொருவருடைய சம்பள விவரத்தையும் பார்த்துத் தனித்தனியே ஒவ்வொருவருக்கும் சம்பளப் பட்டியல் தயாரிப் பதில் எந்தச் சிரமும் இல்லை.

அடுத்து வேறு ஓர் உதாரணத்தைப் பார்ப்போம். ஒரு வங்கி (Bank) தன்னிடம் கணக்கு வைத்திருப்போரின் விவரங்களையும் கணக்குகளையும் காந்த நாடாவில் எழுதிக் கொண்டு அதை கம்ப்யூட்டர் மூலமாக கையாள்வதாக வைத்துக் கொள்வோம். இந்த ஏற்பாட்டின்படி ஒவ்வொரு வாடிக்கையாளருக்கும் ஒரு வாடிக்கை எண் கொடுக்கப் பட்டிருக்கும்.

ஒவ்வொரு முறையும் வாடிக்கையாளர் பணம் போடும் போதும், ஒரு கார்டில் அவரது வாடிக்கை எண்ணையும் போடப் பட்ட தொகையையும் துளைத்து கம்ப்யூட்டருக்கு கொடுத்தால் அது காந்த நாடாவை முதலிலிருந்து கரந்துக்கொண்டு குறிப் பிட்ட வாடிக்கை எண் வந்ததும் அவ்வெண்ணுக்கு உரிய



படம் 3.12

காந்தத் தட்டுகள்

விவரங்களைக் கம்ப்யூட்டரின் நினைவகத்திற்குக் கொண்டுவருகிறது. இவ்வாறு கொண்டுவரப்பட்ட பழைய விவரங்களுடன் இப்போது அவர் நமது கணக்கில் போட்ட தொகையையும் சேர்த்துக் கணக்கை நேர் செய்து மீண்டும் இவ் விவரங்களை காந்த நாடாவில் எழுதிக் கொள்ளுகிறது. இந்த முறையில் நாள் தோறும் நடைபெறும் வங்கியின் அலுவலில், வாடிக்கையாளர்கள் அவர்களது வாடிக்கை எண் வரிசைப்படி வருவார்கள் என்று எதிர் பார்க்க முடியாது; ஆகையால் காந்த நாடாவை பயன்படுத்தி மேற்சொன்ன அலுவல்களைக் கம்ப்யூட்டர் செய்ய வேண்டுமாயின் ஒரு கணக்கு முடிந்ததும் நாடாவை முழுதும் சுற்றிக் கொண்டு அடுத்த வாடிக்கை எண் வரும் போது மீண்டும் முதலிலிருந்து முன்போல் கரந்துக் கொண்டு வந்த அந்த எண் வந்ததும் அதற்குரிய விவரங்களைக் கம்ப்யூட்டரின் நினைவகத்தில் கொண்டு சேர்க்க வேண்டும். இம்முறையில் கால விரயம் அதிகம்.

காந்தத் தட்டு (Magnetic Disk)

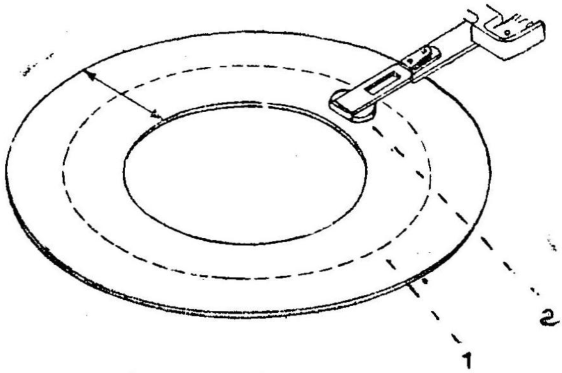
காந்த நாடாவைப் போல, காந்தத் தட்டு மற்றொரு கொள்/கொடு துறையாகும். காந்த நாடாவைப் போலவே, காந்தத் தட்டிலும் விவரங்களைக் காந்தப் புள்ளிகளைப் பயன்படுத்தி எழுதி வைத்துக் கொள்ளலாம். அவ்வாறு எழுதி வைக்கப் பட்டதைக் கம்ப்யூட்டரைக் கொண்டு படிக்கவும் செய்யலாம்.

காந்தத் தட்டு ஒவ்வொன்றும் பார்வைக்குச் சாதாரண கிராம போன் இசைத் தட்டு போலக் காணப்படும். கிராம போன் தட்டில் பதிந்துள்ள இசை, நாம் பார்க்கத் தகுந்த வகையில், தொடர்ந்த கீறல் வடிவில் இருக்கும். ஆனால் காந்தத் தட்டில் செய்திகள் நம் கண்ணுக்குத் தெரியாத காந்தப் புள்ளிகளாய் (ஒருவகையில் காந்த நாடாவில் உள்ளனவைப் போல்) பைனரி முறையில் இருக்கின்றன. அதுவுமல்லாமல் காந்தத் தட்டுக்கள் தனித்தனியாக இருக்காது. பலத் தட்டுக்கள் சேர்ந்த கொத்துக் களாகவே அவை தயாரிக்கப் படுகின்றன. ஒரு கொத்து சாதாரணமாக 6 தட்டுகளைக் கொண்டதாக இருக்கும்.

இந்த 6 தட்டுகளும் ஒரு இணைப்புத் தட்டுடன் சேர்க்கப் பட்டு ஒரே கொத்தாக ஒரு நிமிடத்திற்கு 1500 சுற்றுக்கள் என்ற கணக்கில் சுழலுகின்றது. தட்டுக்களின் ஆரவாக்கில் (Radial direction) தட்டுக்களின் ஒவ்வொரு பக்கத்திற்கும் ஒன்ருகப் படி/எழுது குறிகள் (Read / Write Heads) அமைக்கப் பட்டுள்ளன. இந்தப் படி/எழுது குறிகளும் ஒரே கொத்தாக,

தட்டின் வெளிச் சுற்றிலிருந்து மையத்தை நோக்கி முன்னும், பின்னும் நகரவல்லவை.

இனி காந்தத் தட்டுகளில் எவ்வாறு விவரங்கள் பதியப் படுகின்றன என்று பார்ப்போம். தட்டுக் கொத்தில் (Disc pack) உள்ள பரப்பு ஓர் - மைய வட்டமான தாரைகளாகப் (Tracks of concentric circles) பிரிக்கப் பட்டிருக்கின்றன. ஒவ்வொரு பரப்பிலும் இத்தகைய 100 ஓர்—மைய வட்ட தாரைகள்



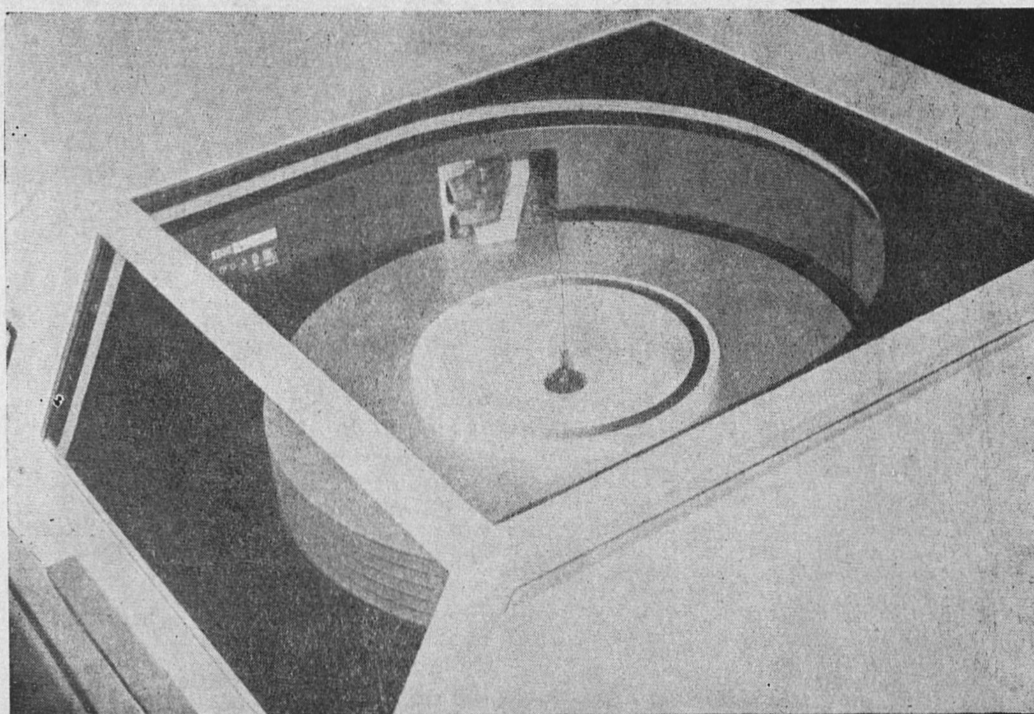
படம் 3.13

காந்தத் தட்டுக் கொத்தின் தட்டுகளில் ஒன்று

1. தாரைகளுள் ஒன்று 2. படி எழுது முனை

உள்ளன. இப்பட்டை ஒவ்வொன்றும் 20 சம துண்டங்களாகப் (Sectors) பிரிக்கப் பட்டுள்ளன. இத்தகைய துண்டம் ஒவ் வொன்றிலும் 100 உருக்களை எழுதுவதற்கான இடம் உள்ளது. இவ்வுருக்கள் ஒவ்வொன்றும் பைனரி முறையில் காந்தப் புள்ளி களாலேயே எழுதப்படுகின்றன.

ஆறு தட்டுக்கள் உள்ள ஒரு கொத்தை எடுத்துக் கொண்டால் அதில் நமக்கு மொத்தம் ஒரு தட்டுக்கு இரண்டு வீதம் பன்னிரண்டு பக்கங்கள் கிடைக்கின்றன. ஆனால் முதல் தட்டின் மேற்புறமும், ஆறுவது தட்டின் அடிப்புறமும் பயன்படுத்தப்படுவதில்லை. தட்டுக் கொத்தை அடிக்கடி மாற்றி வைக்க வேண்டியிருப்பதால், அதன் வெளிப்புறப் பரப்புகள் இரண்டும் கீறல், தூசி படிதல் முதலிய தொல்லைகளுக்கு எளிதில் ஆளாகும். இக்காரணத்தாலும் இன்னும் சில காரணங்களாலும் இவ்விரு புறங்களும் பயன்படுத்தப் படுவ தில்லை. ஆக ஒரு ஆறு தட்டுக் கொத்தில் நமக்கு, மேல், அடிப்



படம் 3.13 a

IBM 1311 காந்தத் தட்டுக் கொத்துகள்—கண்ணுடி மூடி திறந்த நிலை

படம் 3.13b

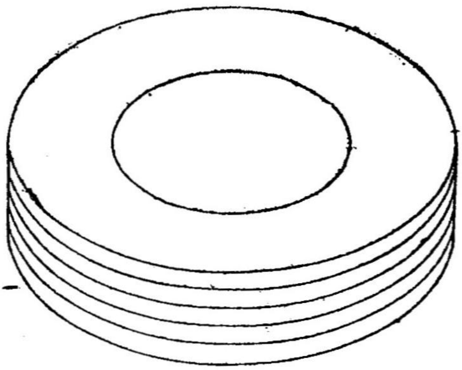
IBM 1311 காந்தத் தட்டுக் கொத்துகள்—கண்ணுடி மூடி மூடிய நிலை

புறப் பரப்புக்கள் நீங்கலாக 10 பரப்புக்கள் கிடைக்கின்றன. ஒவ்வொரு பரப்பிலும் முன் கூறியவாறு 100 ஓர் – மைய வட்டப் பட்டைகளும், ஒவ்வொரு பட்டையிலும் 20 துண்டங்களும் ஒவ்வொரு துண்டத்திலும் 100 உருக்களை எழுத இடமும் உள்ளன. இதன்படிப் பார்த்தால் ஆறு தட்டுக்களைக் கொண்டு உள்ள ஒரு கொத்தில் மொத்தம் 2 மில்லியன் உருக்கள் எழுது வதற்கான இடமும் உள்ளன.

காந்தத் தட்டுக்களிலும் சரி, காந்த நாடாக்களிலும் சரி ஒரு முறை எழுதப்பட்ட செய்தியை நாம் எத்துணை முறை வேண்டுமானாலும் மீண்டும் மீண்டும் 'படிக்கலாம்' ஒரு புத்த கத்தை நாம் படிக்கும் போது அந்தப் புத்தகத்தில் எழுதப்பட்ட எழுத்துக்கள் நாம் படிப்பதால் அழிந்து விடுவதில்லை. எவ்வளவு முறை வேண்டுமானாலும் நாம் அதை திரும்ப திரும்ப படிக்க லாம். இதே முறையில் தான் காந்தத் தட்டு, காந்த நாடா ஆகியவற்றிலும் அவற்றில் எழுதப்பட்டுள்ள செய்திகள், அவற்றை கம்ப்யூட்டர் 'படிப்பதால்' பழுது படுவதில்லை. ஆனால் அதே சமயத்தில் காந்தத் தட்டிலோ அல்லது நாடாவிலோ உள்ள விவரங்கள் நமக்கு தேவையில்லையாயின் அவ்விவரங்கள் உள்ள இடங்களில் அவ்விடத்தில் புதிய விவரங்களை எழுதினால் பழைய விவரங்கள் தாமாகவே அழிக்கப்பட்டு விடுகின்றன. காந்த நாடாவில் எழுத அல்லது படிக்க வேண்டிய விவரங்கள் யாவும் ஒன்றன் பின் ஒன்றாக வரிசை முறையில் (Serial Order) தான் படிக்க / எழுத இயலும் என்று பார்த்தோம். காந்தத் தட்டைப் பயன் படுத்துவதில் என்ன சௌகரியம் என்று பார்ப் போம்.

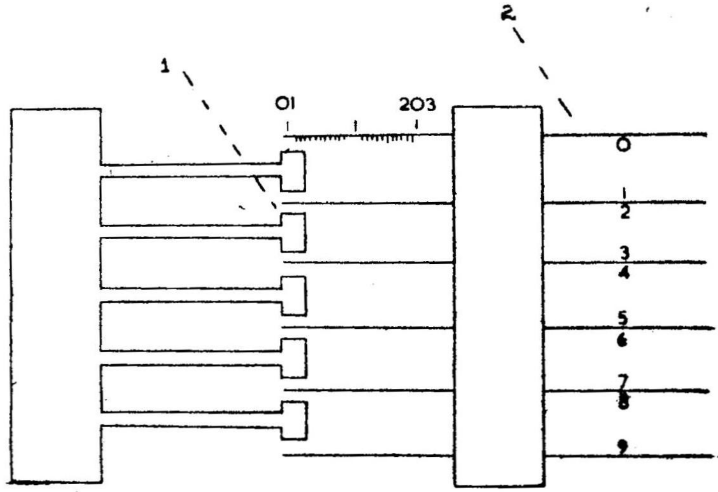
காந்தத் தட்டில் ஒவ்வொரு புறத்திலும் 100 ஓர்- மைய தாரைகள் உள்ளன என்று முன்னரே பார்த்தோம். ஒரு தட்டில் உள்ள ஏதாவது ஒரு தாரையை எடுத்துக் கொண்டால் அதன் நேர் மேலும், நேர் கீழுமாக உள்ள எல்லா தாரைகளும் சேர்ந்து ஒரு உருளை மீது (cylinder) அமைவதாகக் கொள்ளலாம். எடுத்துக் காட்டாக எல்லா தட்டிலும் உள்ள முதல் (தட்டின் விளிம்பு ஓரத்தில் உள்ள) தாரைகளை எடுத்துக் கொள்ளுவோம். இத்தாரைகள் யாவும் சேர்ந்து ஓர் உருளை (உருளை எண்- 1) மீது அமைகின்றன.

இதே முறையில் இரண்டாம் தாரைகளும் மற்றொரு உருளை (உருளை எண்-2) மீது அமையும். இவ்வமைப்பின் படி, உருளை 2, உருளை 1க்குள் அமைந்திருக்கும். இவவாறே மற்றுமுள்ள

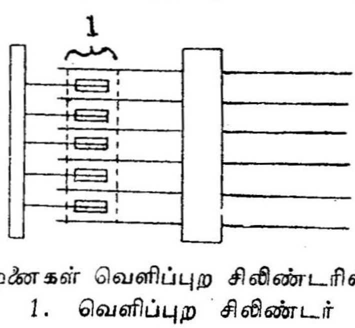


காந்தக் தட்டுக் கொத்து

1. படி எழுது முனைகள் 2. தட்டுக்கள்



தட்டுக் கொத்தின் குறுக்கு வெட்டு தோற்றம்



படி-எழுது முனைகள் வெளிப்புற சிலிண்டரில் நிற்கின்றன

1. வெளிப்புற சிலிண்டர்

படம் 3.14

3, 4, 5 முதலிய தாரைகளும் ஒன்றுக்குள் ஒன்ருக அமைந்த உருளைகள் மீது இருப்பதாகக் கற்பனை செய்து கொள்ளலாம். இக் கற்பனை உருளை ஒவ்வொன்றிற்கும் ஒரு எண் கொடுக்கப் பட்டு, 0-உருளை, 1-உருளை, 2-உருளை 99-உருளை என்று இந்த நூறு உருளைகளும் அழைக்கப் படுகின்றன.

ஒரு உருளை எண்ணும், தட்டின் பக்க- எண்ணும் (6 தட்டுக் கொத்தில் 10 பக்கங்கள் உள்ளன என்பதை நினைவு கூர்க) அத்தட்டின் பக்கத்தில் துண்ட - எண்ணும் (Sector Number) கொடுக்கப் பட்டால், இவை யாவும் சேர்ந்து தட்டுக் கொத்தில் உள்ள துண்டம் ஒன்றின் முகவரியாகிறது.

தட்டுக் கொத்திலுள்ள படி / எழுது குறியில் ஒவ்வொரு தட்டின் ஒவ்வொரு புறத்திற்குமாக ஒரு படி / எழுது முனை உள்ளது. ஆகவே இத்தகைய படி / எழுது முனை ஒன்று தட்டுக் கொத்தின் விளிம்பிலிருந்து ஆரவாக்கில் (Radial direction) மையத்தை நோக்கி உள்புறமாகச் செல்லும் போது, முதலில் வெளிப்புற உருளை மீதும் (உருளை-0) பின்னர் அதற்கடுத்த உருளை மீதும் (உருளை-1) அதன் பின்னர் உருளை 2 என்று ஒவ்வொரு உள் உருளையாக உள் நோக்கிச் செல்லும். எடுத்துக் காட்டாக உருளை எண் 25-ல் படி (எழுது) முனை தற்போது இருப்பதாகக் கொள்ளுவோம். இப்போது தட்டுக் கொத்தின் 25-வது ஓர்— மைய தாரை, ஒவ்வொன்றிற்கும் எதிராக ஒரு படி (எழுது) முனை இடம் பெற்றிருக்கும். படி (எழுது) முனை தட்டைத் தொட்டுக்கொண்டு இருக்காது. தட்டிற்கும் முனைக்கும் இடையே மிகச் சிறு இடை வெளியிருக்கும். இவ்விடைவெளியிலுள்ள காற்று மெத்தை (air cushion) :வழியே தான் படி (எழுது) முனை தட்டின் மீது காந்தப் புள்ளிகளை எழுதவோ அழிக்கவோ செய்கிறது. இப்போது 6-வது படி முனையை மட்டும் செயல்படுத்தினுல் (activate) 25-வது உருளையில் உள்ள மேலிருந்து 6-வது தாரை (அதாவது மூன்ருவது தட்டின் அடிப்புறத்தில் உள்ள 25-வது தாரை) எதிர்ப்படும். தட்டுக் கொத்தில் உள்ள ஒவ்வொரு துண்டத்திலும் அத்துண்டத்திற்கான முகவரி முன்னரே எழுதப் பட்டிருக்கும். ஆகவே குறிப்பிட்ட தாரை எதிர்ப்பட்டதும், அத்தாரையிலுள்ள துண்டங்களின் மீது எழுதப் பட்டுள்ள முகவரியைப் படித்து உரிய துண்டத்தை படி (எழுது) முனை தேர்ந்தெடுக்கிறது. படிக்கும் பொழுது ஒரு துண்டம் தேர்ந்தெடுக்கப் பட்ட உடன், அத்துண்டத்தில் உள்ள விவரங்கள் முற்றுமாக கம்ப்யூட்டரின் நினைவகத்திற்கு கொண்டுவரப் படுகின்றன. அதே போல் எழுதும் போதும் ஒரு முழு துண்டத்

தில் எழுத விரும்பும் செய்தி யாவற்றையும் நாம் ஒன்ருகக் கொடுத்தாக வேண்டும். தட்டுக் கொத்தின் படி (எழுது) முறையில் 'துண்டம்' ஒரு அடிப்படை அளவு, அதாவது ஒரு யூனிட் ஆகும்.

இதுவரை நாம் பார்த்ததிலிருந்து ஒன்று புலனுகிறது. தட்டின் உள்புறமாகவோ அல்லது வெளிப் புறமாகவோ (எங்கிருப்பினும் சரி) உள்ள செய்தியைப் படிப்பதற்கு, முதலிலிருந்து ஒவ்வொரு துண்டமாக நாம் படித்துக் கொண்டு வர வேண்டியதில்லை. ஒரு துண்டத்தின் முகவரி கொடுக்கப்பட்டிருப்பின், நேராக அத்துண்டத்திற்கு படி (எழுது) முனை நகர்ந்து சென்று உரிய விவரத்தை உடனே கொணர முடிகிறது. இக்காரணத்தால், தட்டில் எந்த இடத்தில் உள்ள செய்தியை வேண்டுமானுலும் நேரிடையாக நாம் பெற இயலும். இம்முறையில் காந்த நாடாவில் ஏற்படக் கூடிய கால விரையம் தவிர்க்கப்படுகிறது. இத்தகைய முறையில் விவரங்களை எழுதுவதையும் படிப்பதையும் கட்டற்ற கொள் முறை (Random access Method) என்று கூறுவர்.

முன் கூறிய வங்கியின் கொள்வினை கொடுப்பினைக் கணக்குகளைத் கையாள காந்தத் தட்டுகளைப் பயன் படுத்தலாம். ஒவ்வொரு வாடிக்கையாளருக்கும் ஒரு வாடிக்கை எண் அல்லது கணக்கு எண் (account number) கொடுத்து அந்த எண்ணுக்கும் அது எழுதப்பட்டுள்ள காந்தத் தட்டின் துண்ட முகவரிக்கும் தொடர்பு ஏற்படுத்திக் கொள்ளலாம். (எடுத்துக் காட்டாக ஒரு துண்ட முகவரியையோ வாடிக்கையாளரின் வாடிக்கை எண்ணுகப் பயன்படுத்தலாம்). துண்ட முகவரியை விளிக்கும் போது அதிலிள்ள வாடிக்கையாளரின் எண்ணும் மற்ற விவரங்களும் கம்ப்யூட்டருக்கு உடனே கிடைக்கின்றன. (எவ்வாறு கம்ப்யூட்டரின் நினைவகத்தில் உள்ள எண்களைப் பெற. ஆளங்கம் (control unit) அவ்வெண்களுக்குறிய நினைவறைகளின் முகவரியை விளிக்க வேண்டுமோ, அதே போல் காந்த தட்டில் உள்ள ஏதாவது ஒரு விவரம் தேவைப்படும்போது, அவ்விவரம் எழுதப்பட்டுள்ள துண்டத்தின் முகவரியைக் கம்ப்யூட்டர் விளித்தாக வேண்டும்.)

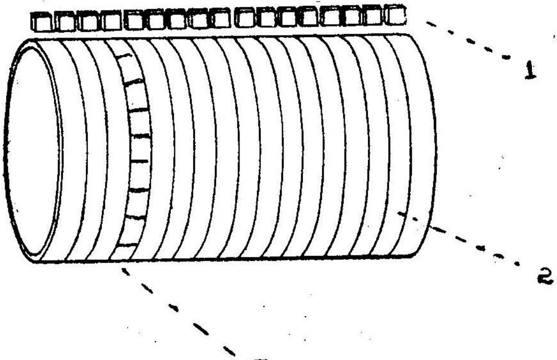
காந்தத் தட்டின் துண்ட முகவரி ஒரு இலக்க எண்ணுகும். ஒரு 6 - தட்டுக் கொத்தில் 20,000 துண்டங்கள் உள்ளன. அவற்றின் முகவரிகளும் 00000 முதல் 19999 முடிய உள்ள ஐந்திலக்க எண்களாகும். இந்த எண்களையே வாடிக்கை எண்ணுகப் பயன் படுத்தலாம். எடுத்துக் காட்டாக வாடிக்கை எண்

17532-க்கு உரிய கணக்கு விவரங்களை முகவரி 17532-ஐ உடைய துண்டத்தில் எழுதி வைத்துக் கொள்ளலாம். இவ் வாடிக்கையாளர் ஒரு முறை ரூ 2000/ வங்கியில் செலுத்துகிருர் என்று வைத்துக் கொள்ளுவோம். அப்போது அவரது கணக்கில் இத்தொகையை வரவு வைத்து, அவரது இருப்பை நாளது வரை நேர்செய்ய வேண்டும் (up to date). ஒரு கார்டில் வாடிக்கை எண் ணையும் (i. e. 17532) செலுத்திய தொகையையும் துளையிட்டுக் கம்ப்யூட்டருக்கு அளிக்கலாம். நாம் பயன்படுத்தும் புரோகிராம் நேரிடையாக துண்ட முகவரி 17532 ஐ விளித்து அங்குள்ள விவரங்களைக் கம்ப்யூட்டரின் நினைவகத்திற்குக் கொணரும். கொணர்ந்த பின்னர் அவரது கையிருப்புத் தொகையுடன் இப் போது கொடுத்த ரூ 2000/=யும் சேர்த்துக்கொண்டு ஒரு புதிய கையிருப்புத் தொகையைக் கணித்து வைத்துக்கொள்ளுகிறது. இது முடிந்ததும் நமது புரோகிராம் துண்ட முகவரி 17532-ல் புதிய கையிருப்புத் தொகையை எழுதுமாறு கப்ப்யூட்டருக்குப் பணிக் கிறது. உடனே புதிய விவரங்கள் காந்தத் தட்டின் 17532 முகவரி உள்ள துண்டத்தில் பதியப் படுகின்றன. இவ்வாருக இந்தக் கணக்கு நேர் செய்யப்படுகிறது. இந்த முறையில் நாம் முக்கிய மாகக் கவனிக்க வேண்டியது, துண்டங்கள் ஒவ் வொன்றையும் நாம் வரிசைக்கிரமாக நமக்கு வேண்டிய துண்டம் வரும் வரை படித்துக் கொண்டே போக வேண்டிய தேவை இல்லாமலிருப்பதாகும். (காந்த நாடாவில் படிக்கும் முறைக்கும் காந்தத் தட்டில் எழுது / படி முறைக்கும் இது ஒரு அடிப்படை யான வேறுபாடு என்பதை அறிக.)

விவரங்களை சேமித்து வைத்துக் கொள்ள காந்த நாடாவும் காந்தத் தட்டும் சிறந்த சாதனங்களாம்; இன்று இவையிரண் டும் பெருமளவிலும் பயன் படுத்தப்பட்டும் வருகின்றன. இவை மட்டுமில்லாமல் வேறு சில சாதனங்களும் விவரங்களைச் சேமித்து வைக்கப் பயன்படுகின்றன. அவற்றுள் காந்த உருளை (Magnetic Drum), டேட்டா செல் (Data cell) ஆகிய இரண்டையும் நோக்குவோம்.

காந்த உருளை.

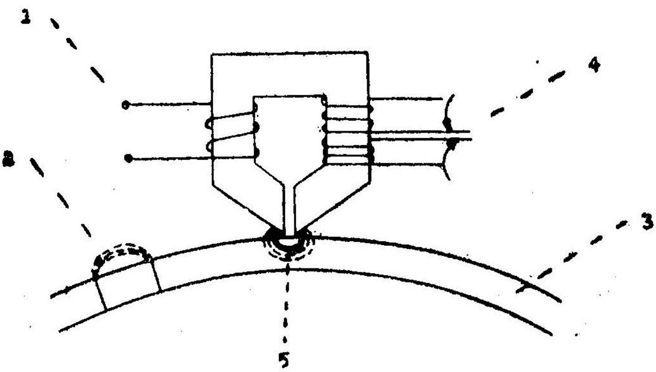
இது ஒரு சிலிண்டர் வடிவ உருளையாகும். இதன் மேற் பரப்பில் உருளையைச்சுற்றி தாரைகள் உள்ளன; படம் 3.14ஐக் காண்க. இத்தாரைகள் மீது இருமை முறையில் (பைனரி) விவரங்களை எழுத ஏற்றவாறு சிறு சிறு காந்த ஏற்புப் புள்ளிகள் உள்ளன. இப்புள்ளிகளை 'பிட்'களாகப் பயன்படுத்தி படித்தலும்,



படம் 3.14 a

காந்த உருளை (Magnetic Drum)

1. காந்த உருளையின் படி-எழுது முகங்கள் (Read Write Head)



3.14 b

காந்த உருளையில் விவரங்கள் எழுதப்படுகின்றன

படி-எழுதுமுகமும் காந்த உருளையின் மேற்பறப்பும்.

1. படி-எழுது முனையில் உள்ள 'படிக்கும்' மின் வழி
2. காந்த உருளையின்மீது எழுதப்பட்டுள்ள ஒரு விவரம்
3. காந்த உருளையின் மேற்பரப்பு
4. படி-எழுது முகத்தின் 'எழுதும்' மின் வழி
5. காந்தக் களம்

எழுதுதலும் நடைபெறுகின்றன. படம் 3.14a-ல் காந்த உருளையின் அமைப்பும் அதன் பல வேறு தாரைகளுக்கான படி-எழுது முனைகளும் காட்டப் பட்டுள்ளன. ஒவ்வொரு தாரைக்கும் மேலே படி / எழுது முனைகள் இருப்பதை நோக்குக.

படித்தலும், எழுதுதலும், காந்த நாடா, காந்தத்தட்டு, காந்த உருளை ஆகிய யாவற்றிலும் அடிப்படையில் ஒரே முறை தான் பின்பற்றப் படுகின்றது.

படம் 3·14b-ல் காந்த உருளை மீது படிக்க / எழுத பயன் படுத்தப்படும் படி / எழுது முனையின் அமைப்பு காட்டப் பட்டுள்ளது. மின் காந்தச் சட்டம் ஒன்றின் இரண்டு புயங்களில் பக்கத்திற்கு ஒன்ருக இரண்டு வகை பின்வழிகள் (1, 4, என்று படத்தில் குறிக்கப் பட்டிருப்பவை) உள்ளன. அவற்றுள் '4' 'எழுதும்' பின் வழியாகும். எழுத வேண்டிய உருவுக்கான (Character) பைனரிக் குறிகள் மின் துடிப்புகள் உருவில் இந்த மின் கம்பிகளில் செலுத்தப்படும். இத்துடிப்பு ஒவ்வொன்றும் ஒரு சிறு மின் காந்தப்புலனை (Magnetic field) ஏற்படுத்துகிறது. இக்காந்தப் புலன் நிலையானதல்ல. மின் துடிப்பு மறைந்ததும் இப்புலனும் மறைந்துவிடும். காந்தப் புலன் இருக்கும் போது அதன் நேர் கீழே, காந்த உருளையின் மேற்பரப்பில் உள்ள தாரை ஒன்று இருக்கும். இத்தாரையில் உள்ள காந்தமேற்கும் புள்ளிகளில் ஒன்று (Magnetisable Spot) அதன் அருகே வர அப் புள்ளிகள் காந்த மேற்றப்படுகிறது. காந்தப் புள்ளியில் ஏற்றப் பட்ட காந்தம் நிலையானதாகும். இவ்வாறு, காந்த உருளையில் எழுதும் போது, ஒவ்வொரு மின் துடிப்பும் ஒரு காந்தப் புள்ளி யாக மாற்றப்பட்டு வேண்டிய விவரங்கள் பைனரி முறையில் பதிந்து கொள்ளப் படுகின்றன. காந்த உருளையிலிருந்து விவரங் களைப் படிக்கும் போது இதற்கு நேர் மாருன நிகழ்ச்சி ஏற்படு கிறது. 'படிப்பு' நிகழும் போது காந்தப்புள்ளிகள் படி / எழுது முனைக்குக் கீழே வரும் போது 'படிக்கும்' மின் வழியில் (1- என்று குறிப்பிட்டுள்ள மின் வழி) மின் துடிப்பை ஏற்படுத்துகிறது. இத் துடிப்பின் திண்மையைப் பிரிதோரிடத்தில் அதிகரித்து, கம்ப் யூட்டரின் நினைவகத்திற்கு அனுப்பப்பட்டு அங்கே மீண்டும், அது காந்த உருவில் பதிந்து கொள்ளப்படுகிறது.

ஒரு காந்த உருளையின் ஒவ்வொரு தாரையும் பல துண்டங் களாகப் பிரிக்கப் பட்டுள்ளது. ஒரு குறிப்பிட்டக் கம்ப்யூட்ட ரைச் சார்ந்த காந்த உருளையில் (Royal-precision-LGP-30) 64 தாரைகளும், ஒவ்வொரு தாரையிலும் 64 துண்டங்களும் உள்ளன. இந்த $64 \times 64 = 4096$ துண்டங்கள் ஒவ்வொன்றிலும் 32 — 'பிட்'களை எழுத இயலும். அதாவது துண்டம் ஒவ்வொன் றிலும் 32 காந்த மேற்புப் புள்ளிகள் உள்ளன.

சில ஆண்டுகளுக்கு முன் வரை காந்த உருளைகள் கம்ப் யூட்டரின் நினைவகங்களாகவும் பயன் படுத்தப் பட்டு வந்தன

அவற்றைக் காட்டிலும் சிறந்த காந்த வளைய நினைவகம் (Core Momory) வந்த பின்னர் காந்த உருளைகளை முக்கிய நினைவகங்களாகப் பயன் படுத்துவது வழக்கொழிந்து போயிற்று.

டேட்டா செல் (Data Cell)

இதுவரை நாம் பார்த்த சாதனங்கள் யாவற்றையும் விட கொள்ளளவில் **(Capacity)** மிகப் பெரியது டேட்டா செல் ஆகும். இதில் சிலிண்டர் வடிவில் 10 அறைகள் அடுத்தடுத்து உள்ளன. ஒவ்வொரு அறையிலும் 20 சிற்றறைகள் உள்ளன. ஒவ்வொரு சிற்றறையிலும் 10 பட்டைகளும் ஒவ்வொரு பட்டையிலும் 100 தாரைகளும் உள்ளன. இந்தத் தாரைகள் ஒவ்வொன்றும் ஒரு காந்த நாடாத் துண்டு போல் காணப்படும். இந்தத் தாரைகள் ஒவ்வொன்றிலும் 4000 தசம, இலக்கங்களை (Decimal digits) எழுதிக் கொள்ளுவதற்கான இடம் உள்ளது. ஆக ஒரு டேட்டா செல்லின் கொள்ளளவு:

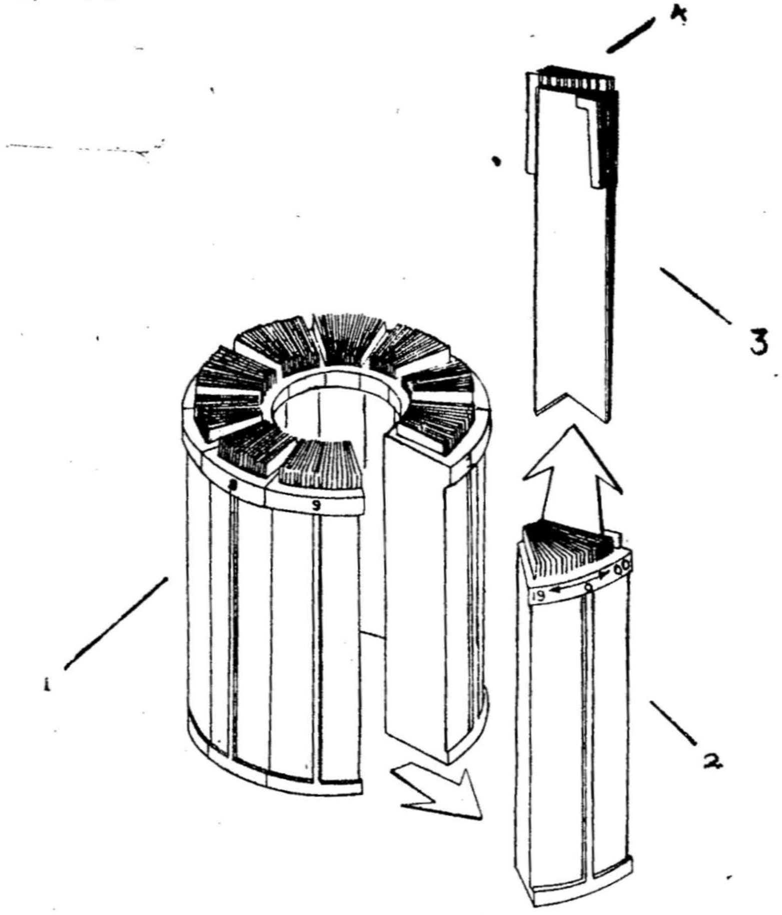
$4000 \times 100 \times 10 \times 20 \times 10 = 800,000,000$ (800 மில்லியன்) தசம இலக்கங்களாம். இது, ஒவ்வொன்றிலும் 100 தாரைகளைக் கொண்ட 2000 காந்த உருளைகள், அல்லது 1000 பக்கங்களும், ப்க்கத்திற்கு 40 வரிகளும், வரிக்கு 50 உருக்களும் கொண்ட 200 புத்தகங்களுக்குச் சமம்.

ஒரு விவரத்தை டேட்டா செல்லில் கம்ப்யூட்டரைக் கொண்டு எழுதும் போது, உரிய செல்லும், அதிலுள்ள குறிப்பிட்ட சிற்றறையும், சிற்றறையில் உள்ள பட்டையும் தேர்ந்தெடுத்துக் கொள்ளப் படுகின்றன. பிறகு இந்தப் பட்டை தானகவே, சிற்றறையிலிருந்து வெளிவந்து ஒரு உருளை மீது சுற்றிக் கொள்கிறது; சுற்றி முடிந்ததும் இவ் வுருளை சுழல, அதனருகில் உள்ள படி (எழுது) முனை படிக்க (எழுத) வேண்டிய பணியைச் செய்கிறது. இப்பணி முடிந்ததும், பட்டை தானகவே மீண்டும் தனது இடத்திற்குச் சென்று விடுகிறது. டேட்டா செல்லில் பட்டையைத் தேர்ந்தெடுத்து உருளையில் சுற்றி எழுத / படிக்கத் தயாரிக்க ஆகும் நேரம் ஏறத்தாழ ஒரு நிமிடம் ஆகும். ஆனல் சுற்றி முடிக்கப்பட்ட பின்னர் நொடிக்கு 110,000 தசம இலக்கங்கள் என்ற வீதத்தில் டேட்டா செல்லிலிருந்து விவரங்களைக் கம்ப்யூட்டர் படிக்க இயலும்.

நேர்முக அச்சுப்பொறி (on line printer)

இன்றைய **அசுர வேகக்** கம்ப்யூட்டர்களில் பயன் படுத்தப் படும் கொடங்கங்களில் நேர்முக அச்சுப் பொறி முதன்மையான

தாகும். இப்பொறி கம்ப்யூட்டரின் தலைமை ஆய்வுப் பொறியுடன் நேரடியாக இணைக்கப்பட்டிருக்கும். இவ்வச்சுப் பொறியில் ஒரு நகரும் சங்கிலி உள்ளது. இச்சங்கிலியின் மீது எழுத்துக்களும் எண்களும் இன்னும் பிற குறிகளும் சங்கிலி முழுதும் நிறையுமாறு அமைந்துள்ளன. இந்த சங்கிலியின் பின்புறம் சிறு சிறு மின்

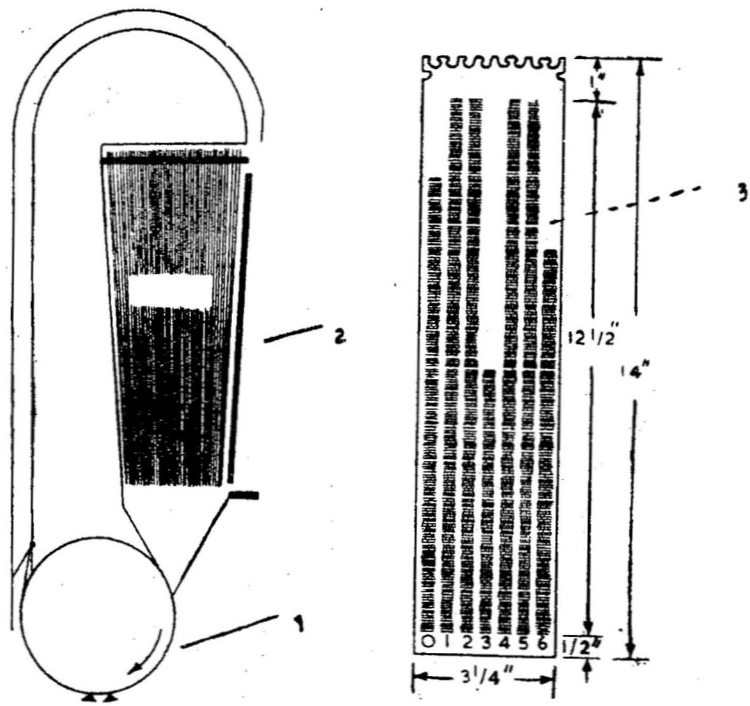


படம் 3.14 c

1. டேட்டா செல் உருளை 2. ஒரு டேட்டா அறை
3. ஒரு டேட்டா சிற்றறை 4. டேட்டா பட்டைகள்

காந்தச் சம்மட்டிகள் உள்ளன. சம்மட்டிகளுக்கும் சங்கிலிக்கும் இடையே அச்சிடப்பட வேண்டிய தாள் செருகப்பட்டிருக்கும். சங்கிலி சுழலும் போது அச்சிடப் படவேண்டிய எழுத்துக்களுக்கு

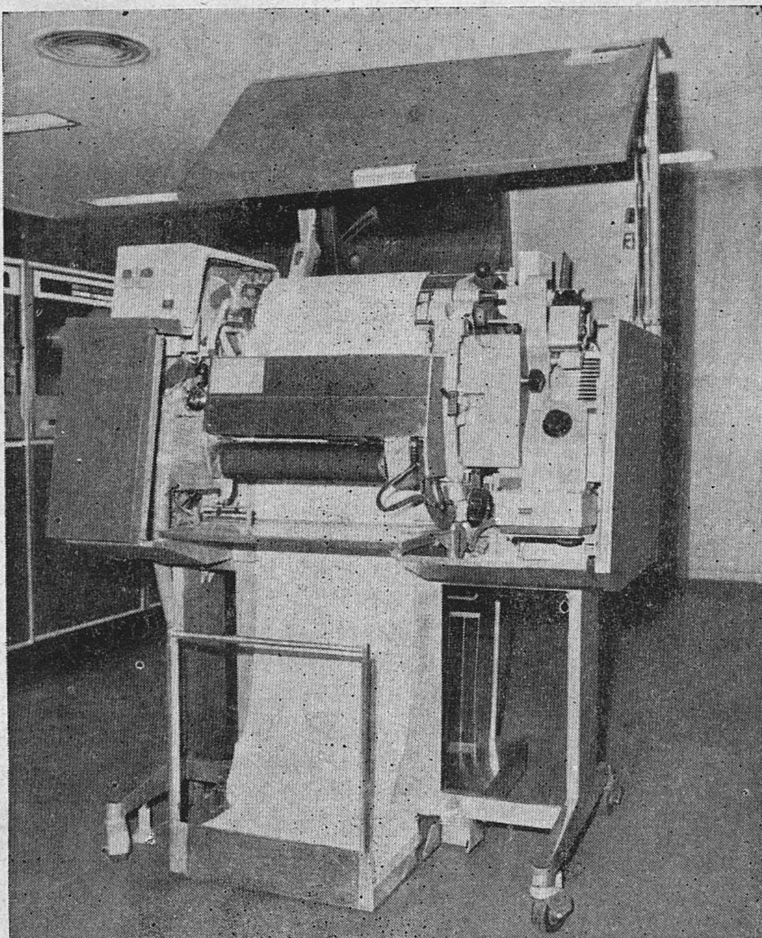
உரிய சம்மட்டிகள் மட்டும் மின் காந்தத்தால் இழுக்கப்பட்டு அவ்வெழுத்துக்கள் மீது மோதும். அவ்வாறு மோதும் போது இடையே உள்ள காகிதத்தில் அவ்வெழுத்துக்கள் பொறிக்கப்படும். இத்தகைய அச்சுப் பொறிகளில் ஒரு வரியில் 132 எழுத்துக்கள் வரை பொறிக்க இயலும்; ஒரு நிமிடத்திற்கு (வரிக்கு 132 எழுத்துக்கள் கொண்ட) 1200 வரிகளை இப்பொறிகள் அச்சிட வல்லன; வரிக்கு 40 எழுத்துக்களும், பக்கத்திற்கு 40 வரிகளையும் கொண்ட புத்தகத்தின் 92 பக்கங்களை அச்சிட இப்பொறி ஒரு நிமிடமே எடுத்துக் கொள்ளுகிறது. இருப்பினும் இது குறைந்த வேகப் பொறியே.



படம் 3.14 d

டேட்டா செல் பட்டையும், பட்டை சுற்றப்படும் உருளையும்

1. டேட்டா செல் சுற்றப்படும் உருளை
2. உருளையில் சுற்றப்பட காத்திருக்கும் பட்டைகளின் தொகுதி
3. பட்டையில் எழுதப்பட்டுள்ள விவரங்களின் (காந்தக் குறி) அமைப்பு



படம் 3.15

IBM 1403—நேர்முக அச்சுப் பொறி

4

ஒற்றைப் பதிவுப் பொறிகள்

(Unit Record Machines)

முன்னொரு அத்தியாயத்தில் கம்ப்யூட்டர்களில் பயன் படுத்தப் படும் துளைக் கார்டுகளின் வரலாற்றைப் பார்த்தோம். இத்துளைக் கார்டுகளைக் கையாளும் பொறிகள் சிலவற்றை ஹெர்மன் ஹாலரித், ஜேம்ஸ் பவர்ஸ் போன்றோர் தயாரித் தார்கள் என்றும் கண்டோம். இவர்கள் தயாரித்த பொறிகளின் சந்ததிகளாக இன்று ஒரு கூட்டமே உள்ளது. இக் கூட்டத்தில் உள்ள பொறிகளை 'ஒற்றைப் பதிவுப் பொறிகள்' (Unit Record Machines) என்பர்.

சாதாரணமாக நாம் கையாளும் விவரங்களைப் பெருவகை யாக (Broadly) கோவைகளாகப் (Files) பிரிக்கலாம். எடுத்துக் காட்டாக ஒரு தொழிற்சாலையில் வேலை செய்யும் தொழிலாளர் களின் சம்பள விவரங்கள் மொத்தத்தையும் சேர்த்து ஒரு கோவை (சம்பளக் கோவை) எனக் குறிப்பிடலாம். அந்தத் தொழிற்சாலையில் தயாராகும் பொருள்கள் பற்றிய தொகுப்பைப் பிறிதொரு கோவையாகக் கருதலாம். இவ்விதம் வகைப் படுத்தப்பட்ட விவரங்களைக் கொண்ட ஒவ்வொரு தொகுப்பை யும் ஒரு கோவை என்று கூறலாம்.

மேற் கூறிய கோவைகள் ஒவ்வொன்றிலும் பல பதிவுகள் (Records) அடங்கியுள்ளன. எடுத்துக் காட்டாக, சம்பளக் கோவயை எடுத்துக் கொள்வோம். இந்தக் கோவையில் ஏதா வது ஒரு தொழிலாளியின் சம்பள விவரத்தை மட்டும் எடுத் தால் அது ஒரு பதிவு (Record) ஆகும். ஆக ஒரு கோவை, ஒரே வகைப்பட்ட பதிவுகளின் தொகுதியாகும். இத்தகைய ஒரு பதிவை நாம் கார்டில் துளைக்கிறோம் என்று வைத்துக் கொள்ளுவோம். சில சமயங்களில் பதிவுகள் சிறியவையாக இருப்பின் ஒரு கார்டுக்கு ஒரு பதிவு என்ற விதத்தில் நாம்

பதிவுகளைக் கார்டுகளில் துளைத்துக் கொள்ள முடியும்; பிற சமயங் களில் ஒரு பதிவுக்கு ஒன்றிற்கு மேற்பட்ட கார்டுகள் தேவைப் படலாம்.

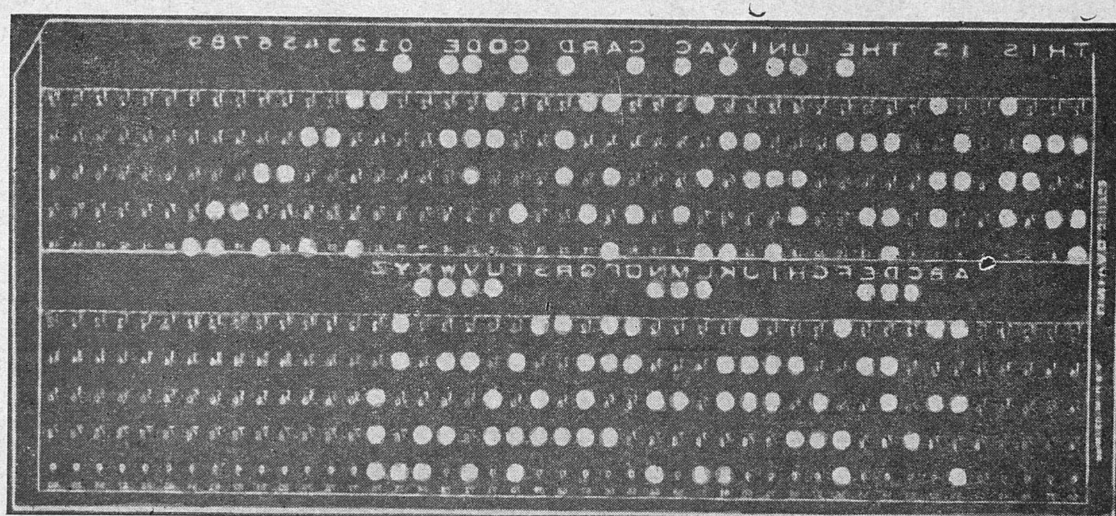
ஒரு முழுக் கார்டில் ஒரு பதிவு பற்றிக் காணப்படும் விவரங்கள் (அது முழுப்பதிவாயிருப்பினும் சரி, அல்லது ஒரு பெரிய பதிப்பின் பகுதியாயிருப்பினும் சரி அதை 'யூனிட் ரெக்கார்டு' (Unit Record) அல்லது 'ஒற்றைப் பதிவு' என்று கூறுவர்.

அத்தியாயத்தின் முற்பகுதியில் குறிப்பிடப்பட்டுள்ள 'ஒற் றைப் பதிவுப் பொறிகள்' யாவும் மேற்சொன்ன ஒற்றைப் பதிவு களைக்கையாள வல்லவை. ஒற்றைப் பதிவுகளைக் கொண்டே சில எளிய கணிப்புகளைச் செய்து முடித்துவிட முடியும்; கம்ப் யூட்டரின் உதவி, இத்தகைய கணிப்புகளுக்குச் சிறிதும் தேவை யிருக்காது. சில வகை ஆய்வுகளில் ஒரு பகுதியை ஒற்றைப் பதிவுப் பொறிகளில் செய்து முடித்த பின்னர், மற்ற பகுதியைக் கம்ப்யூட்டரில் செய்வது, சிக்கனமானதாகவும், குறைந்த நேரத் தில் முடியக் கூடியதாகவும் அமைவதுண்டு. இவ்வத்தியாயத் தில் இத்தகைய பொறிகள் சிலவற்றைப் பற்றி மேலெழுந்த வாரி யாகப் பார்க்கலாம். அதற்கு முன்னராக, இப்பொறிகள் கை யாளும் கார்டு வகைகளைப் பற்றிப் பார்ப்போம்.

கார்டு வகைகள்

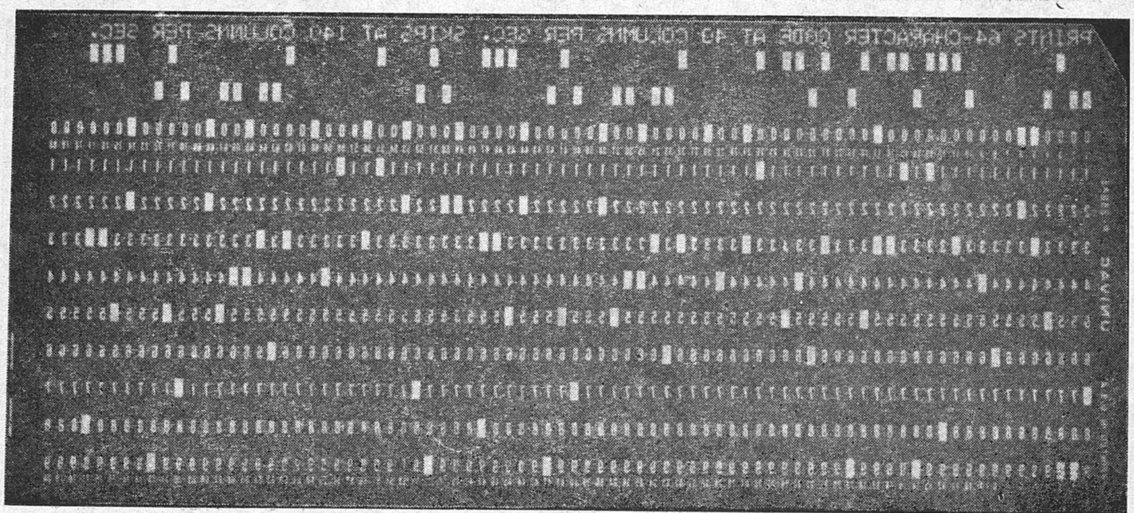
ரெமிங்டன் ரேண்ட் கார்டு

கம்ப்யூட்டர் பயன் படுத்தும் கார்டின் அமைப்பைப்பற்றியும், அதன் மீது இடப்படும் துளைக்குறிகள் பற்றியும் ஏற்கனவே பார்த் தோம். நாம் பார்த்த குறியீடுகள் யாவும் ஹாலரித் குறியீடுகள் ஆகும். கம்ப்யூட்டர் உலகின் பெரும் பகுதி இந்த ஹாலரித் குறியீட்டைத் தான் பயன்படுத்துகிறது. ஹாலரித் குறியீட்டைப் தவிர வேறொரு குறியீட்டைப் பயன் படுத்தும் கார்டுகளும் உண்டு. அவை ரெமிங்டன் ரேண்ட் கார்டுகளாம். ஐ. பி. எம். கார்டும் (ஹாலரித்) ரெமிங்டன் ரேண்ட் கார்டும் நீளம், அகலம், பருமன் ஆகியவற்றில் ஒத்திருந்தாலும் அவற்றிடையே பல வேறு பாடுகள் உண்டு. ரேண்ட் கார்டுகளில் உள்ள துளைக் குறியீட்டு அமைப்பு ஐ. பி. எம். (ஹாலரித்) கார்டின் அமைப்பில் இருந்து முற்றிலும் வேறுபட்டிருக்கும். ஐ. பி. எம் கார்டுகளில் இடப்படும் துளைகள் செவ்வக வடிவமானவை; ரெமிங்டன் கார்டின் துளைகள் வட்ட வடிவமானவை. ஐ. பி. எம். கார்டில், இலக்கங்களைக்



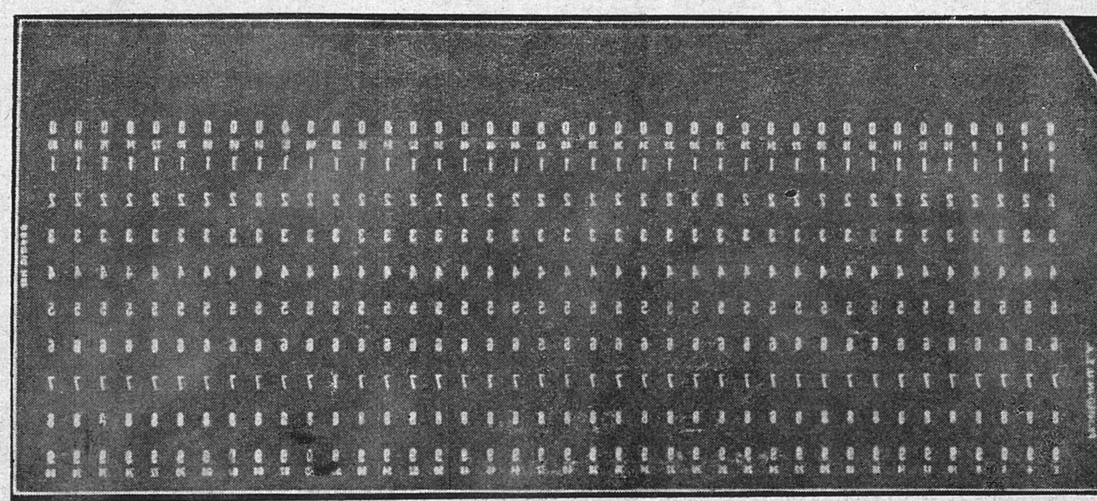
படம் 4.1

ரெமிங்டன் ரேண்ட் கார்டின் அமைப்பு



படம் 4.1a

துளைக்கப்பட்ட ஹாலரித் கார்டு



படம் 4.2

உடனடி—துளைக் கார்டு (Port-A Punch Card)

குறிக்க (0 முதல் 9 ஐ குறிக்க) அந்த குறிப்பிட்ட இலக்கத்தின் மீது துளையிடுவோம். அதாவது ஒரு இலக்கத்திற்கு ஒரு துளை என்ற குறியீட்டு அமைப்பைக் கைக் கொள்வோம். ஆனால் 'ரெமிங்டன் ரேண்ட்' துளைக் குறியீட்டில் ஒற்றைப் படை இலக்கத்திற்கு ஒரு துளைக் குறியீடும், இரட்டைப்படை இலக்கத் திற்கு இரு துளைக் குறியீடும் பயன்படுத்தப்படுகின்றன. எழுத்துக் களிலும் கூட (A முதல் Z வரை) அவற்றிற்கு ஒரே சீரான, துளையீடு கிடையாது. சிலவற்றிற்கு இரட்டை துளைகளும் (B,E,K,L etc) மற்றவற்றிற்கு மூன்று துளைகளும் (A,D,J...) உப யோகப்படுத்தப்படுகின்றன. இதுவல்லாமல் ரெமிங்டன் ரேண்ட் கார்டு நீளவாக்கில் இரண்டு தனிப் பகுதிகளாகப் பிரிக்கப் பட்டுள்ளது. மேல் பகுதியில் ஒரு தொகுதி வரிசையும் (zone) ஐந்து இலக்க வரிசைகளும் உள்ளன. இந்த ஆறு வரிசைகளும் கார்டின் கீழ்ப் பகுதியில் மீண்டும் காணப்படுகின்றன. இத்தகைய அடிப்படை வேறு பாடுகளால், ஐ. பி. எம் கார்டுகளைப் பயன்படுத்தும் பொறிகள் ரெமிங்டன் கார்டு களையோ, அல்லது ரெமிங்டன் கார்டுகளைப் பயன் படுத்தும் பொறிகள் ஐ. பி. எம் கார்டுகளையோ பயன் படுத்த இயலாது. இவ்விரண்டு வகைக் கார்டுக் குறிகளும் பொருந்தாத தன்மையன. (Incompatible). கம்ப்யூட்டர் உலகில் ரெமிங்டன் ரேண்ட் கார்டு களைப் பெருமளவில் பயன் படுத்துவதில்லை. அதனால் பின் வரும் பக்கங்களில் நாம் முற்றும் ஐ. பி. எம் கார்டுகளைப் பற்றியும் அவற்றைக் கையாளும் பொறிகளைப் பற்றியுமே பார்ப்போம்.

உடனடி — துளைக்கார்டு (Port — A — Punch Card)

சாதாரணமாக துளைக் கார்டுகளில் துளைக்கப்பட வேண்டிய செய்திகளை முதலில், சாதாரணக் காகிதங்களில் எழுதிக்கொண்டு பின்னர், துளைப் பொறிகளைப் (Punching Machines) பயன்படுத் திக் கார்டுகளில் துளைக்கிறார்கள். ஆனால் சில சமயங்களில் இத்தகைய ஏற்பாடு பயன்படாது. செய்திகள் அல்லது விவ ரங்கள் கிடைத்து, பின்னர் அவற்றை எழுக்கொண்டு, பிறிதோரி டத்திற்கு அக்காகிதத்தை அனுப்பி, அதிலுள்ள விவரங்களைத் துளைத்து அதன் பின்னர் அந்த கார்டுகளை கம்ப்யூட்டர்களில் பயன்படுத்தி ஆய்வதற்கு (Process) ஏற்படக்கூடிய கால இடை வெளி சில சமயங்களில் மிகவும் அசௌகரியமானதாக இருக்க லாம். எடுத்துக்காட்டாக ஒரு தவணை முறை வியாபாரி தனது வாடிக்கையாளரிடம் தினந்தோறும் தவணைகளை வசூல் செய்வ தாக வைத்துக்கொள்வோம். ஒவ்வொரு வாடிக்கையாளருக்கும்

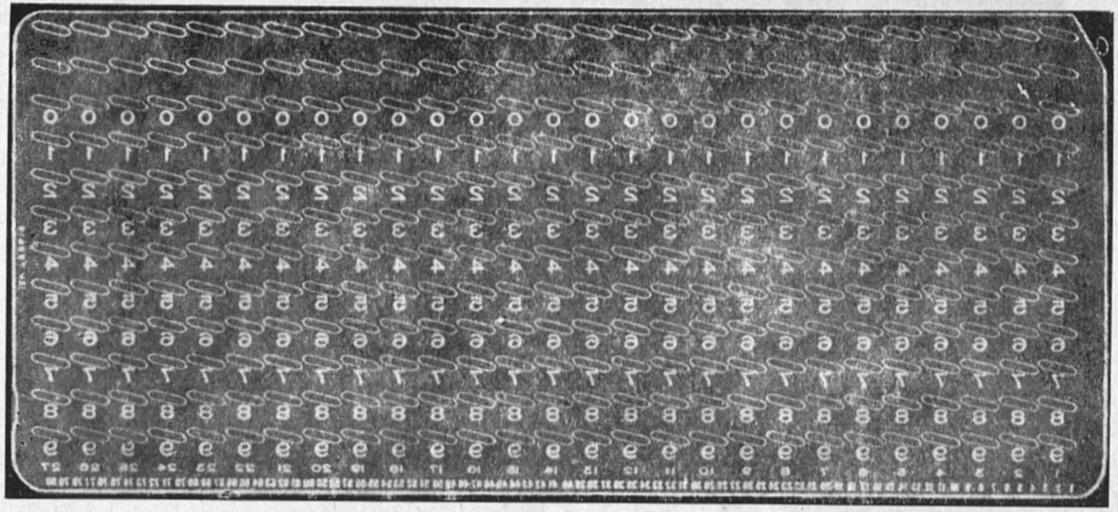
ஒரு வாடிக்கை எண் அளிக்கப்பட்டு வாடிக்கையாளரின் கணக்குகள் முழுதும் ஒரு கம்ப்யூட்டர் பார்த்துக்கொள்வதாக வைத்துக்கொள்வோம். சாதாரண முறைப்படிக் கணக்கைக் கையாள்வதாயிருப்பின் வாடிக்கையாளர் பணம் செலுத்தச் செலுத்த அவ்விவரங்களைக் குறித்துக்கொண்டு அவற்றை துளைப் பொறி அறைக்கு (Punch room) அனுப்புவார்கள். அங்கே அவ்விவரங்களைக் கார்டுகளில் துளைத்து, சரிபார்த்து துளைக்கப்பட்ட கார்டுகளைக் கம்ப்யூட்டர் அறைக்கு அனுப்புவார்கள். இம் முறையில் ஏற்படக்கூடிய காலச் செலவை முன்னிட்டு வேறொரு முறை கையாளப்படுகிறது.

இத்தகைய துறைகளில் சாதாரணமாகப் பயன்படுத்தும் கார்டுகளுக்குப் பதிலாக ஒருவகைச் சிறப்புக் கார்டுகளைப் பயன்படுத்தலாம். இக்கார்டுகளை 'உடனடி துளைக் கார்டு' (Port A Punch) என்று அழைப்போம். இக்கார்டுகளில் ஒரு பகுதி மற்ற சாதாரணக் கார்டுகளைப் போலவே இருக்கும். மற்ற பகுதியில் கார்டின்மீது அச்சிடப்பட்டுள்ள இலக்கங்களைச் சுற்றி செவ்வக வடிவில் மயிர்க்கண் துளைகள் (Perforations) அமைந்திருக்கும். ஒரு கூறிய முனையுள்ள கம்பியைக்கொண்டு இவ்விலக்கத்தை அழுந்தினால், உடனே அவ்விலக்கம் துளைத்தெடுக்கப்பட்டு அங்கு துளைப் பொறியில் இருப்பதைப்போன்ற ஒரு செவ்வகத் துளை ஒன்று உண்டாகும்.

மேற்சொன்ன எடுத்துக்காட்டில், ஒரு வாடிக்கையாளர் வந்து பணம் செலுத்தியவுடன், ஒரு உடனடித் துளைக் கார்டில் உடனே அவரது வாடிக்கை எண்ணையும் செலுத்தியத் தொகையையும், நாம் துளைத்து விட்டால் அக்கார்டை உடனே கம்ப்யூட்டர் அறைக்கு ஆய்வதற்காக (Process) அனுப்பலாம். இதனால் சாதாரண முறையில் ஏற்படக்கூடிய கால விரயம் தவிர்க்கப்படுவதுடன் பணச் செலவும் குறைகிறது; கார்டுகளில் துளையிடுவதற்காகத் தனிப்பட்ட துளைப் பொறிகளும் தேவையில்லை.

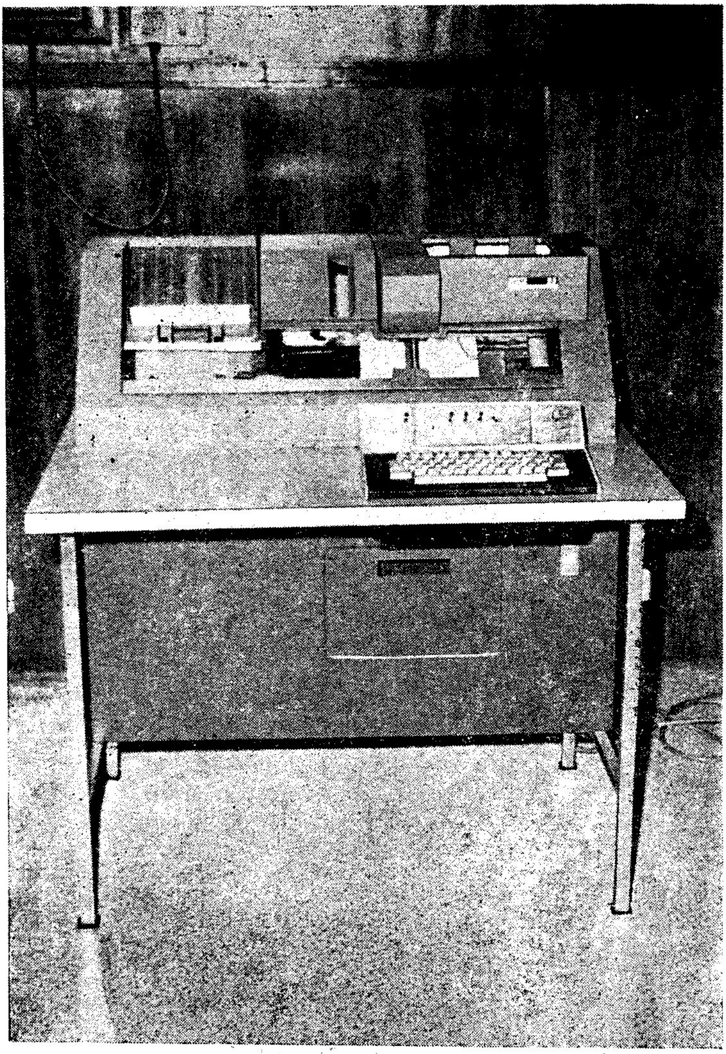
குறியுணர் கார்டு (Mark - Sensed Cards)

உடனடித் துளைக் கார்டு போலவே அமைந்த மற்றொரு சாதனம் குறியுணர் கார்டு. இந்தக் கார்டிலும் ஒரு பகுதி சாதாரண கார்டில் உள்ளதுபோலவே இருக்கும். இந்தக் கார்டின் மற்றொரு பகுதிக்கு குறியுணர் பகுதி என்று பெயர். இந்தப் பகுதியும், முதல் பகுதியைப் போலவே குத்து நெடு வரிசைகளைக் கொண்டதாய் இருக்கும்.



படம 4.3

குறியுணர் கார்டு



படம் 4.4

துளைப் பொறி (Punching Machine)

இப்பகுதியில் உள்ள எண்கள்மீது, ஒரு தனிவகைப் பென்சில்கொண்டு குறியிடலாம். படம் 4.3-ல் (0, 1, 2 ஆகியவை குறிப்பிடப்பட்டுள்ளன) இவ்வாறு குறிக்கப்பட்டுள்ள கார்டுகளை, ரிபுரொடியூசர் (Reproducer) என்ற பொறியில் செலுத்தினால், இப்பொறி பென்சிலினால் குறிக்கப்பட்ட பகுதிகளை படித்து அதற்கு இணையான (Corresponding) குத்து, நெடு வரிசையில் முன் பகுதியில் உள்ள இலக்கத்தை துளைத்தெடுக்கும்.

இக் கார்டுகளை உபயோகிப்போர் சற்று கவனமாக இருத்தல் அவசியம். இக் கார்டுகளில் இடப்படும் குறிகள் உரிய இடத்தில் அமைய வேண்டும்; அழுத்தமான ஒற்றைக்கோடு மட்டுமே குறியீட்டுக்குப் பயன்படுத்த வேண்டும். தவறான இடத்தில் குறியிடப்பட்டிருப்பின் அதைத் தெளிவாகக் களைந்துவிடவேண்டும். எக்காரணம் கொண்டும் இடப்பட்ட குறி, உரிய இடத்தை மிஞ்சி அடுத்த இடத்தில் போய்ச் சேரக்கூடாது.

பிற கார்டுகள்

மேற்சொன்ன இரண்டுவகைக் கார்டுகளைத்தவிர இன்னும் பலவிதமான கார்டுகளும், (காந்த கார்டுகள், மைக்ரோ பில்ம் கார்டுகள் போன்றவை) புழக்கத்தில் உள்ளன. அவற்றின் பயனும், அவற்றைப் பயன்படுத்தும் கருவிகளைப் பற்றிய விவரமும் இப் புத்தகத்தின் நோக்கத்திற்கு அப்பாற்பட்டவை.

துளைப் பொறி (Punching Machine)

இதுவரை நாம் சிலவகையான துளைக் கார்டுகளைப் பற்றிப் பார்த்தோம். இப்போது இக் கார்டுகளில் துளையிடும் பொறிகள் பற்றியும், துளைக் கார்டுகளைப் பயன்படுத்தும் யூனிட் ரெக்கார்ட் பொறிகள் பற்றியும் பார்ப்போம்.

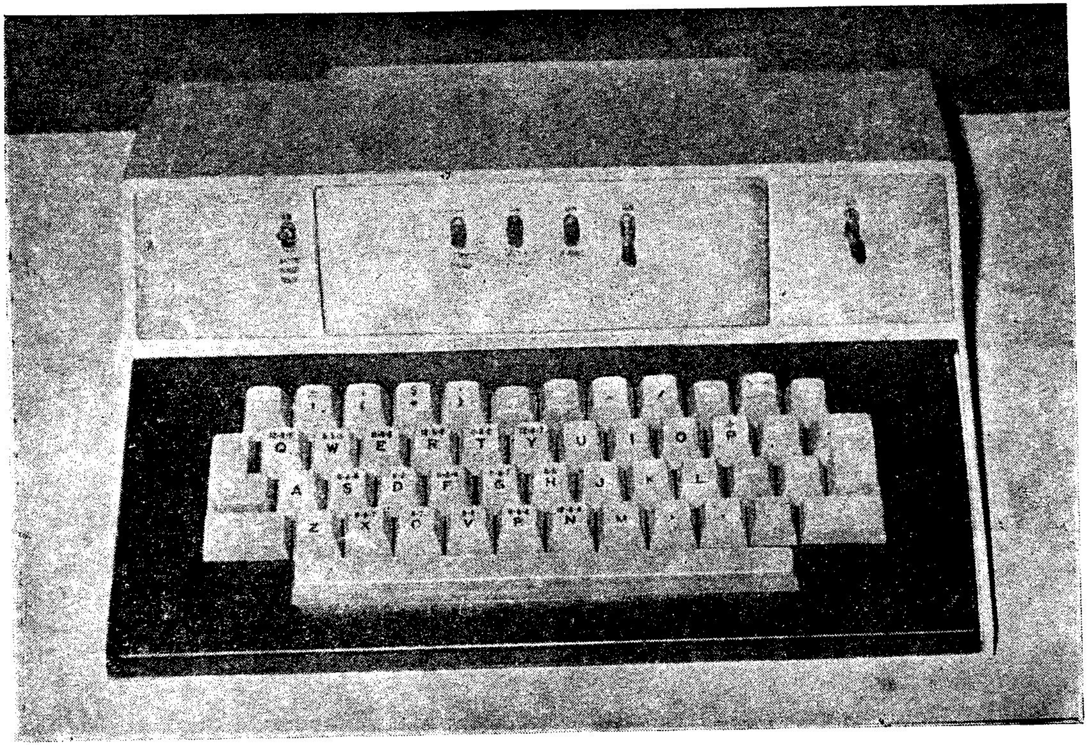
ஐ. பி. எம். கார்டுகளிலாகட்டும், ரெமிங்டன் ரேண்ட் கார்டுகளிலாகட்டும் கம்ப்யூட்டர் பயன்படுத்தும் உருக்கள் (Characters) யாவும் துளைச் சேர்க்கைகளால் குறியிடப்படுகின்றன என்று பார்த்தோம். இத்தகைய துளைகளைப் பொறுக்கி எடுத்து உரிய இடத்தில் (அல்லது இடங்களில்) துளையிடுவதற்கு என சில பொறிகள் அமைந்துள்ளன. இப் பொறிகளை நாம் 'துளைப் பொறிகள்' (Punching Machines) என்போம். படம் 4.4-ல் இத்தகைய துளைப் பொறி ஒன்றின் புகைப்படம் உள்ளது. இப் பொறியின் முக்கிய பாகங்கள் (i) பித்தான் முகம், (ii) கார்டு இடுமிடம், (iii) துளை மேடை, (iv) துளை முகம், (v) இடை மேடை, (vi) படிப்பிடம், (vii) சேர்ப்பு மேடை, (viii) சேர்ப்பிடம், (ix) புரோகிராம் உருளை ஆகும்.

பித்தான் முகம் அமைப்பில் ஒரு தட்டெழுத்துப் பொறியைப்போல் காணப்படுகிறது. படம் 4.5-ஐ காண்க. இதிலுள்ள பித்தான்களில் சிலவற்றின்மீது ஒரே ஒரு எழுத்து (Q, W, E, etc.,) மட்டும் காணப்படுகிறது. இன்னும் சிலவற்றில் ஒரே ஒரு இலக்கம் (8, 9) மட்டும் காணப்படுகிறது. மீதியுள்ள பித்தான்களில் ஒவ்வொன்றிலும் இரண்டு குறியீடுகள் காணப்படுகின்றன. இந்த அமைப்பின் இயக்கமும் சாதாரண தட்டெழுத்துப் பொறியின் இயக்கத்தைப் போன்றதே. இந்த பித்தான் முகம், கீழ் நிலை, மேல் நிலை போன்ற இரண்டு நிலைகளில் இயங்குகிறது. சாதாரணமாக இம்முகம் கீழ் நிலையில் இருக்கும். அப்போது இப் பித்தான்களை அழுத்துவதால் A முதல் Z வரை உள்ள எழுத்துக்களையும், 8, 9 என்ற இலக்கங்களையும், மற்ற பித்தான்களில், கீழ் குறிகளையும் துளை செய்யலாம். எடுத்துக் காட்டாக பித்தான்-முகத்தின் மேல் வரிசையில் இடது புறத்தில் ஒரு பித்தானில் $ என்ற குறியும் அதன் கீழே * என்ற குறியும் காணப்படுகின்றன. கீழ் நிலையில் இப் பித்தானை அழுத்தினால் * குறிக்கான துளைகள் கார்டில் இடப்படும். அதுபோலவே 2, 1 ஆகிய இரு உருக்களையும் கொண்ட பித்தானைக் கீழ் நிலையில் அழுத்தினால் I என்ற எழுத்து கார்டில் துளைக்கப்படும். இவற்றுடன் 8, 9 என்ற இலக்கங்களையும் கீழ் நிலையில் இயக்கி உரிய எண்ணை துளைக்கலாம். இந்த இரண்டு பித்தான்கள் மட்டும் (8, 9) கீழ் நிலை, மேல் நிலை ஆகிய இரண்டு நிலைகளிலுமே இயங்க வல்லன.

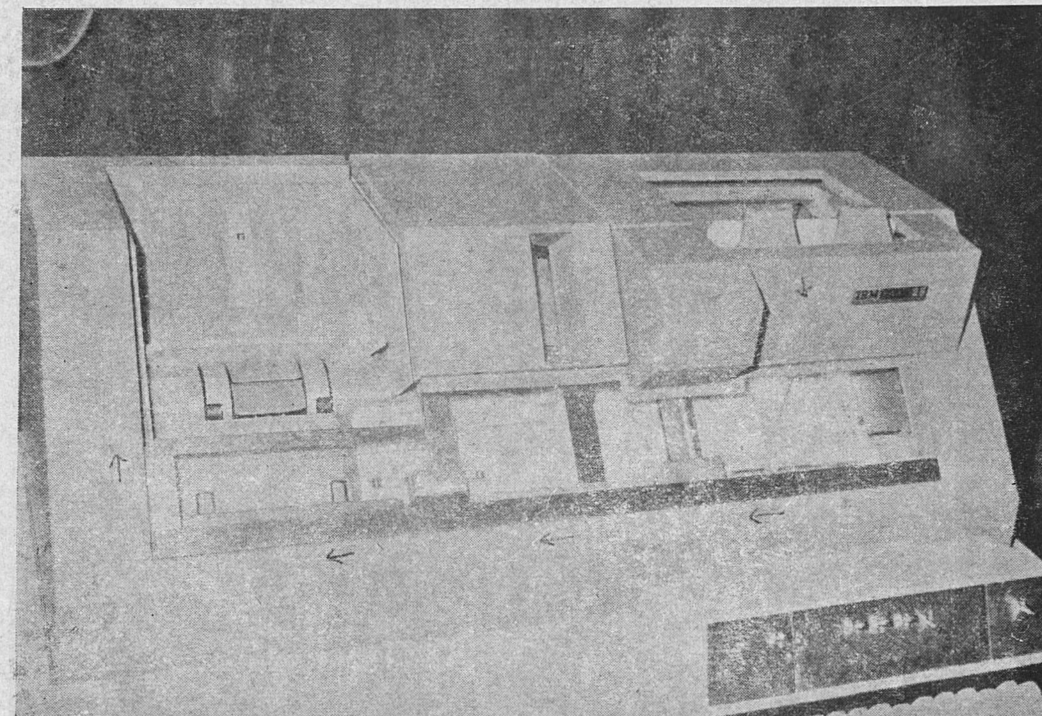
கீழ் நிலையை எழுத்து நிலை (Alphabetic mode) என்றும், மேல் நிலையை (Numeric mode) என்றும் கூறலாம். பித்தான் முகத்தின் கீழ்புறம் இடது கோடியில் 'NUM' என்று ஒரு பித்தான் காணப்படுகிறது. இப் பித்தானை அழுத்திக்கொண்டால் மற்ற பித்தான்கள் யாவும் மேல் நிலையில் மட்டுமே இயங்கும். 'NUM' பித்தானை அழுத்தாமல் மற்ற பித்தான்களை இயக்கும் போது அவை கீழ் நிலையில் இயங்குகின்றன. எழுத்துக்களை மட்டுமே கொண்ட A, S, D போன்ற பித்தான்கள் மேல் நிலையில் இயங்கா.

இப்போது துளைப் பொறியை இயக்கி எவ்வாறு கார்டுகள் துளைக்கப் பெறுகின்றன என்று பார்ப்போம்.

துளையிடப்பட வேண்டிய கார்டுகள் சீராக அடுக்கப்பட்டு, கார்டு இடுமிடத்தில் அடுக்கப் பட வேண்டும். இவ்வாறு அடுக்கப் படும் போது கார்டுக் கற்றையின் மேற் புறம் (அச்சிடப்பட்டுள்ள



படம் 4.5
துளைப் பொறியின் பித்தான் முகம் (Key Board)



படம் 4.5a

IBM 029 கார்டு துளைப் பொறியில் துளைக்கப்பட்ட கார்டுகள் செல்லும் பாதை

பக்கம்) நம்மை நோக்கி இருக்குமாறு அடுக்க வேண்டும். நமது அடுத்த பணி இந்த கார்டுகளை ஒன்றன் பின் ஒன்ருக துளை மேடைக்கு அனுப்பித் துளை முகத்தில் துளைத்து, சேர்ப்பு மேடையில் அடுக்க வேண்டியதாம். இதற்காக பித்தான் முகத்தின் வலப்புறம் அமைந்த சில பித்தான்கள் பயன்படுத்தப்படும்.

'FEED' என்று குறிக்கப்பட்டுள்ள பித்தானை முதலில் அழுத்த வேண்டும். உடனே கார்டிடுமிடத்தில் உள்ளக் கற்றையின் முதல் கார்டு துளை மேடைக்கு அனுப்பப் படுகிறது. துளை மேடைக்குச் சென்றுள்ள கார்டு, துளைப்பதற்கு ஏற்றவாறு இப்போது அமைந்திருப்பதில்லை. கார்டில் துளையிடுவதற்கு முன்னர் கார்டை இடம் திருத்தி வைக்க வேண்டும். இதற்கு என ஒரு தனிப்பட்ட பித்தான் உள்ளது. அதன் மீது 'REG' என்று ('REGISTER' — என்பதன் சுருக்கம்) எழுதப்பட்டிருக்கும்.

'REG' பித்தானைத் தாழ்த்திஞல் துளைமேடைக்குச் செலுத்தப்பட்ட கார்டின் முதல் இடம் (First Column) துளை முகத்தில் படியுமாறு அக்கார்டு இடம் திருத்தி நிறுத்தப்படுகிறது. இப்போது இந்தக் கார்டு துளைகளைப் பெறுவதற்கு ஆயத்தமாக உள்ளது. எடுத்துக்காட்டாக நாம் 'COMPUTER CENTRE' என்ற தொடரை கார்டின் 10-வது இடத்தில் (10th Column) இருந்து துளைக்க வேண்டும் என்று வைத்துக் கொள்ளுவோம். கார்டின் முதல் 9 இடங்களை நாம் காலியாக விட வேண்டும். இதற்கு ஏற்றவாறு தட்டெழுத்துப் பொறியில் உள்ளதைப் போல் 'வெளிச் சட்டம்' (Space Bar) ஒன்று உள்ளது. (பித்தான் முகத்தின் கீழே உள்ள நெடிய செவ்வகச் சட்டம்.) இந்தச் சட்டத்தை ஒவ்வொரு முறை தட்டும் போதும், கார்டில் ஒரு இடம் துளையிடப் படாமல் தாண்டிச் செல்லப்படுகிறது. ஆகவே நமது எடுத்துக் காட்டுக்கு ஏற்றவாறு முதலில் 9 முறை வெளிச்சட்டத்தைத் தட்ட வேண்டும். பின்னர் பத்தாவது இடத்திலிருந்து உரிய பித்தான்களை அழுத்தி (COMPUTER CENTRE) என்ற வார்த்தையைத் துளைக்கிறோம். ஒரு எழுத்தைத் துளைத்தவுடன் தாஞகவே அடுத்த இடம் துளை முகத்கில் அமையுமாறு கார்டு வலமிருந்து இடமாக நகருகிறது. கார்டு இவ்வாறு நகரும் போது, துளைக்கப்பட்ட பகுதி இடை மேடையிலும், துளைக்கப் படவிருக்குமிடம் துளை முகத்திலும் எஞ்சிய பகுதி துளைமேடையிலும் இருக்கும்.

மேற்சொன்ன வார்த்தைத் தொடர், கார்டில் ஒன்றும் இடத்திலிருந்து 23ம் இடம் முடிய உள்ள இடத்தை நிரப்பும். இந்தக்

கார்டில் மீதியுள்ள பகுதியில் நாம் வேறொன்றும் துளைக்கப் போவது இல்லை. இந்தக் கார்டை வெளிக் கொணர்ந்தாக வேண்டும். துளைப்பு வேலை முடிந்தவுடன் (REL; RELEASE) என் பதின் சுருக்கம்) என்ற பித்தானை அழுத்தினால் துளைமேடை யிலிருந்து கார்டு முற்றிலும் விடுபட்டு இடைமேடைக்கு வந்து சேரும். இதற்குப் பின்னர் REG. என்ற பித்தானையும் REL என்ற பித்தானையும் முறையே ஒன்றன்பின் ஒன்றாக அழுத்த, கார்டு இடைமேடையை விட்டு சேர்ப்பு மேடைக்குச் சென்று பின்னர் அங்கிருந்து விடுபட்டு சேர்ப்பிடம் வந்து சேரு கிறது. சில துளைப் பொறிக் கருவிகளில் "CLEAR" என்ற பித்தான் இருக்கும். துளைகள் போட்டவுடன் நாம் வெளியே கொண்டு வர எண்ணினால் "CLEAR" பித்தானை அழுத்தினால் கார்டு சேர்ப்பிடும் இடத்திற்கு வந்துவிடும்.

தட்டெழுத்துப் பொறியில் இருப்பதைப் போல துளைப்பொறி களிலும் 'பின்தள்ளி' (Back Space Key) ஒன்றும் உண்டு. இந்தப் பித்தானை அழுத்திக் கொண்டிருந்தால் துளைமேடையில் சிறிதும், இடைமேடையில் மிகுதியுமாக உள்ள துளைக்கார்டு சிறிது சிறிதாக பின் நோக்கி நகர்ந்து கொண்டேபோய் முற்று மாகத் துளைமேடைக்கு வந்து சேரும்.

டைப்ரைட்டரில் நாம் டைப் செய்ய வேண்டிய இடம் (Position) எதுவென்று நாம் நேரிடையாகப் பார்த்துத் தெரிந்து கொள்ள இயலும். ஆனால் துளைப்பொறியில் அவ்வாறில்லை. துளைப்பொறியின் துளைமுகம் ஒரு குழலால் மூடப்பட்டிருக்கும். அதனால் துளைக்கப்படும் இடமும் அதற்கு இட,வலமாக இரண்டி ரண்டு இடங்களும் இக்குழலால் மறைக்கப் பட்டிருக்கின்றன. இதனால் நேரிடையாகத் துளைக்கப்பட வேண்டிய இடத்தை அறிந்து கொள்ள வேறொரு வசதி செய்து தரப்பட்டிருக்கிறது. கார்டின் அகலத்திற்குச் சமமான உயரமுடைய ஒரு சிறு உருளை சுழலுவதற்கு ஏற்றவாறு ஒரு அச்சில் இணைக்கப்பட்டுள்ளது. இந்த உருளையின் அடி, மேல் ஓரங்களில் 1 முதல் 80 வரை எண்கள் குறிக்கப் பட்டிருக்கும். துளைப் பொறியை இயக்க இயக்க இந்த உருளையும் சுற்றிக் கொண்டே வரும். உருளைக் கருகாமையில் ஒரு நிலையான (Fixed) முள் (Indicator) அமைக்கப்பட்டுள்ளது. உருளைச் சுற்றச் சுற்ற, அதன் மீது குறிக்கப் பட்டுள்ள எண்கள் ஒவ்வொன்றாக அந்த முள்ளுக்கு எதிராக வந்து கொண்டேயிருக்கும். எடுத்துக் காட்டாக நாம் ஒரு கார்டில் 9 இடங்களில் துளையிட்ட பிறகு 10-வது இடம் துளையிடப்பட வேண்டும் எனக் கொள்வோம். அப்போது

உருளையில் '10' என்ற எண், உருளை முள்ளுக்கு எதிரே வந்து நிற்கும். 10-ம் இடத்தைத்துளைத்தவுடன், '11' உருளை—முள்ளுக்கு எதிரே வரும். அதாவது எந்த ஒரு நிலையிலும் உடனடியாகக் கார்டின் எந்த இடத்தில் துளையிடப்பட வேண்டுமோ அந்த இடத்தின் எண் (அல்லது குறி) உருளை-முள்ளுக்கு எதிரே வந்து நிற்கும். இந்த உருளை-முள்ளும் அதன் எதிரே உள்ள உருளையின் சிறு பகுதியும் துளைப்பொறியை இயக்குபவருக்குத் தெரியுமாறு ஒரு சிறு ஜன்னல் அமைந்துள்ளது. இந்த ஜன்னல் வழியாகத் தெரியும் உருளை-முள்ளைப் பார்த்தே நாம் ஒரு கார்டில் எங்கே துளையிடுகிறோம் என்பதை அறிகிறோம். இந்த உருளைக்கு 'புரோகிராம்' உருளை (Program Cylinder) என்று பெயர்.

சில வகைகளில் துளைப் பொறியின் இயக்க முறையும், அமைப்பும் டைப்ரைட்டரை ஒத்திருந்தாலும், அடிப்படையாக பல :வேறுபாடுகள் இவ்வியந்திரங்களுக்கு இடையே உள்ளன. எடுத்துக்காட்டாக டைப்ரைட்டரில் பொறிக்கப்பட்டு முடிந்த செய்திக்கான பிரதி ஒன்றை டைப்ரைட்டரைக் கொண்டே எடுக்கவேண்டுமானால் அதை மீண்டும் டைப் செய்வதைத் தவிர வழியில்லை· ஆனால் துளைப் பொறியில் அவ்வாறில்லை. துளைக்கப்பட்ட கார்டு ஒன்றை இடைமேடையிலும், வெற்றுக் கார்டு ஒன்றைத் துளைமேடையிலும் இருக்குமாறு பொருத்திவிட்டு 'DUP' (DUPLICATE) என்ற பித்தானை அழுத்திக் கொண்டு இருந்தால் இடைமேடையில் உள்ள கார்டில் காணும் எல்லாக் குறியீடுகளும் அப்படியே பிரதியாகத் துளைமேடையில் உள்ள கார்டில் துளைக்கப்படும். ' DUP '-பித்தானை அழுத்திக் கொண்டு இருக்கும் போது இரண்டு மேடைகளிலும் உள்ள கார்டுகள் ஒருசேர நகருகின்றன. இடைமேடைக்கும் சேர்ப்பு மேடைக்கும் இடையே படிப்பிடம் (Reading Station) ஒன்று இருக்கிறது. DUP - பித்தானை அழுத்திக் கொண்டு இருக்கும் போது, இடைமேடையில் உள்ள கார்டு, இப் படிப்பிடத்தைக் கடந்து சேர்ப்பு மேடைக்குச் செல்லுகிறது. இடைமேடைக் கார்டின் எந்த இடம் படிப்பு மேடையில் உள்ளதோ, துளைமேடைக் கார்டின் அதே இடம் துளைமுகத்தின் மேல் இருக்கும். எடுத்துக்காட்டாக இடைமேடைக் கார்டின் 7-வது இடம் படிப்பிடத்தின் மேலே உள்ளது என்று வைத்துக்கொள்ளுவோம். அப்போது துளைமேடைக் கார்டின் 7-வது இடம் துளை முகத்தின் மேல் அமர்ந்திருக்கும். இடைமேடைக் கார்டு 7-வது இடத்தை விட்டு 8-வது இடத்திற்கு நகருகிறது. இவ்வாறு நகரும்போது படிப்பிடம் தனது வழியாகச் செல்லும் கார்டின் துளைகளைப் படித்து அதைத்

துளைப்பிடத்திற்கு உடனே அனுப்புகிறது. துளைப்பிடம் உடனே படிப்பிடத்தில் படிக்கப்பட்டதை அப்படியே துளைமேடைக் கார்டில் துளைக்கிறது. அதன்பின்னரே கார்டு அடுத்த இடத்திற்கு நகருகிறது. எடுத்துக் காட்டாக இடைமேடைக் கார்டில் 7-வது இடத்தில் 3 என்ற எண்மீது துளையிடப்பட்டுள்ளது என்று வைத்துக் கொள்ளுவோம். இந்தக் கார்டின் 7-ம் இடம் படிப்பிடத்தின் மேல் செல்லும்போது இந்தத் துளை படிக்கப்பட்டு, அவ்விவரம் துளை மேடைக்கு அனுப்பப்படுகிறது. அந்நேரத்தில் துளைமேடைமீது, அங்குள்ள கார்டின் 7-ஆம் இடம் பொருந்தியிருக்கும். இவ்விடத்தில் படிப்பிடத்தில் இருந்து வந்த விவரத்தின் பேரில் '3' என்ற எண் இப்போது துளைக்கப்படுகிறது. இவ்வாறே இடைமேடைக் கார்டில் காணும் துளைச் செய்தி (Punched Message) முழுதும் துளைமேடைமீது நகரும் கார்டில் துளைக்கப்பட்டு, இடைமேடைக் கார்டின் பிரதி ஒன்று நமக்குக்கிடைக்கிறது. இம் முறையில் ஒரு கார்டின் முழுப் பிரதியையுமோ, அல்லது பகுதியை மட்டுமோ, மற்றொரு கார்டில் பிரதியாக (Copy) தயாரிக்க இயலும்.

சற்று முன்னர் புரோகிராம் - உருளையைப் பற்றிச் சிறிது அறிந்தோம். இந்த உருளைக்கு வேறு ஒரு முக்கியமான பணியும் உண்டு. ஏற்கெனவே கூறியவாறு இந்த உருளையின் உயரம் ஒரு கார்டின் அகலத்திற்குச்சமமாக இருக்கிறது. உருளையின் சுற்றளவு ஒரு கார்டின் நீளத்திற்குச் சமமாகும். அதனால் ஒரு முழுக் கார்டை இந்த உருளையின் மீது கச்சிதமாகச் சுற்றி வைக்க முடியும். இவ்வாறு சுற்றப்படும் கார்டுக்கு 'புரோகிராம் கார்டு' என்று பெயர். ஒரு புரோகிராம் கார்டில் குறிப்பிட்ட வகைத் துளைகளையிட்டு, அதை உருளையில் சுற்றி வைத்து துளைப் பொறியை இயக்குவதன் மூலம், துளைப் பொறியை வெவ்வேறு நிலைகளில் இயக்க முடியும். சாதாரணமாக நமது தட்டெழுத்துப் பொறியில், அட்டவணைகளைத் தயாரிக்கும் போது, வசதிக்காக இடம் ஒதுக்கல் முறையைக் (Space Setting) கையாளுகிறோம். இம்முறைப்படி அட்டவணையில் உள்ள விவரணம் (Detail) ஒவ்வொன்றிற்கும் ஒரு பகுதியை ஒதுக்கிக் கொள்ளுகிறோம். இப்பகுதிகளுக்கு ஏற்றவாறு தட்டெழுத்துப் பொறியின் பின்புறத்தில் உள்ள சட்டம் ஒன்றில் சில மாறுதல்களை (Space Setting) நாம் செய்கிறோம்.

இத்தகைய இடம் ஒதுக்கல் முறையை நாம் துளைப்பொறிகளிலும் கையாளலாம். இதற்கு உதவியாக நாம் முன்னர் குறிப்பிட்ட உருளைக் கார்டு — அல்லது புரோகிராம் கார்டு பயன்

படுகிறது. மீண்டும் ஒரு எடுத்துக் காட்டைப் பார்ப்போம். ஒரு குறிப்பிட்ட கணிப்பிற்காக 10,000 கார்டுகளை நாம் துளைக்க வேண்டியிருக்கிறது எனக் கொள்ளுவோம். ஒவ்வொரு கார்டிலும் முதல் 5 இடங்களிலும் 5 இலக்க எண்கள் (தொழிலாளியின் 'தொழிலாளி எண்' எனக் கொள்க.) துளைக்கப்பட வேண்டும். பின்னர் 26 முதல் 30 முடிய வெற்றிடமாக விடப்பட்டு, 31 லிருந்து 37 முடிய தேதி (18-7-63 என்ற மாதிரியில்) துளையிடப்பட வேண்டும் எனக் கொள்ளுவோம். இந்த 10,000 கார்டுகளிலும் தேதிமட்டும் ஒன்றாகவே இருக்கும். மற்ற விவரங்கள் பொதுவாக மாறித்தான் காணப்படும். சாதாரண முறையில் இந்தக் கார்டுகளைத் துளைக்க வேண்டுமாயின், முதலில் NUM பித்தானை அழுத்திக் கொண்டு, கார்டின் முதல் ஐந்து இடங்களில் துளையிட்ட பின்னர், அப்பித்தானை விட்டுவிட்டு 6 முதல் 25 இடம் வரை துளைக்கவேண்டும். 26-லிருந்து 30 இடம் முடிய வெளிச்சட்டத்தைத் தட்டிக் கொண்டே போய்ப்பின்னர் மீண்டும் NUM பித்தானை அழுத்திக் கொண்டு தேதியைத் துளையிட வேண்டும். ஒன்றிரண்டு கார்டுகளாயின் இம்முறை பெரும்பாலும் போதுமானதே. ஆனால் ஆயிரக்கணக்கான கார்டுகளைத் துளைக்க வேண்டியிருக்கும் போது, பித்தான் மாற்றல், வெளிச்சட்டத்தைத் தட்டல் ஆகிய செயல்களுக்கு ஆகும் மொத்த நேரம் மிகவும் அதிகமாகும். இத்தகைய கால விரயத்தைத்தடுக்க, துளைப் பொறியில் ஓர் வசதியுள்ளது. உருளைக் கார்டில் உரிய துளைகளை இட்டு, அதன் கட்டுப்பாட்டிற்குட்பட்டு கார்டுகளை துளைக்கும் போது, எண் - நிலைக்கு மாறல் (Change Over to Numerical Mode), அதிலிருந்து, எழுத்து- நிலைக்கு மாறல் (Change Over to Alphabetic Mode) இடைவெளி விடுதல் ஆகிய இயக்கங்கள் தாமாகவே நடைபெறும்படிச் செய்யலாம்; இதற்கென வேறு எந்தப் பித்தானையும் அழுத்தவோ ஸ்விட்ச்சை மாற்றவோ வேண்டியதில்லை. பிறிதொரு எடுத்துக் காட்டைக் காண்போம். ஒரு வெற்றுக் கார்டுக் கற்றையில் ஒவ்வொரு கார்டிலும் முதல் 9 இடங்களை வெற்றிடமாக விட்டு விட்டு, 10 முதல் 19 வரை உள்ள இடங்களில் ஒரு 10 இலக்க எண்ணையும் 20 முதல் 25 முடிய இடத்தை வெற்றிடமாகவும், 25 முதல் 44 முடிய உள்ள பகுதியில், எழுத்துக்களையும் துளைக்க வேண்டும் என்று கொள்ளுவோம். சாதாரண முறையில் இதைச் செய்ய வேண்டுமாயின், ஒரு கார்டைத் துளைமேடைக்குக் கொணர்ந்ததும், 9 வரை வெளிச்சட்டத்தை (Space Bar) தட்டவேண்டும். தட்டி முடிந்ததும் துளைமேடைக் கார்டின் பத்தாம் இடம் துளை முகத்தின் மீது இருக்கும்.

இப்போது நாம் 10 முதல் 19-வரை உள்ள இடங்களில் எண்களைத் துளைக்கப் போகிறோம். ஆகவே, துளைப் பொறியின் பித்தான் முகத்தில் உள்ள NUM என்ற பித்தானை அழுத்திக் கொண்டு, பித்தான் முகத்திலுள்ள எண்-பித்தான்களை (Numeric Keys) இயக்குகிறோம். அதன் பின்னர், ஐந்து முறை (20 முதல் 24 முடிய) வெளிச்சட்டத்தைத் தட்ட, கார்டின் 25-ம் இடம் துளை முகத்தின் மீது அமைகிறது. இங்கிருந்து 44-வது இடம் முடிய நாம் பெயரைத் துளைக்க வேண்டும். இதற்கு, NUM பித்தானை விட்டுவிட்டு பித்தான் முகத்தில் உள்ள A முதல் Z ஈறாக உள்ள எழுத்துகளைப் பயன்படுத்தி பெயர்களைத் துளைக்கிறோம். இவ்வாறு, கற்றையிலுள்ள ஒவ்வொரு கார்டுக்கும் வெளிச்சட்டத்தை தட்டுதல், எண்-நிலைப் பித்தானை (Num Key) அழுத்துதல், பின்னர் எழுத்துக்களைத் துளைக்க ஆரம்பிக்கு முன்னர் விட்டு விடுதல் ஆகியவற்றைத் திருப்பித் திருப்பி செய்ய வேண்டியிருக்கிறது.

இதற்குமாருக, புரோகிராம் கார்டு ஒன்றைப் பயன்படுத்தி இவ்வேலையை எளிதாகச் செய்யலாம். ஒரு வெற்றுக் கார்டில் கீழ்வரும் துளைகளை இடுவோம்.

நெட்டு வரிசை எண் (Column Number)	துளையிடப்படும் உரு (Character Punched)
1	—
2-9	+
10	துளை ஏதும் இல்லை
11-19	+
20	—
21-24	+
25	1
26-44	A
45	—
46-80	+

மேற் கண்டவாறு துளையிடப்பட்ட கார்டை ஒரு உருளைக் கார்டாக (Drum Card), துளைப்பொறியின் புரோக்கிராம் உருளை மீது சுற்றி வைத்து, இந்தக் கார்டின் கட்டுப் பாட்டில் வெற்றுக் கார்டுகளை துளையிட ஆரம்பிப்போம். நமது புரோக்கிராம் கார்டு என்ன சொல்லுகிறது? புரோக்கிராம் கார்டில் ஒரு இடத்தில் '—' குறியும் அடுத்துவரும் இடங்களில் + குறியும் துளைக்கப் பட்டிருப்பின் இதற்கு பொருள் '—' குறியும் அதை அடுத்து

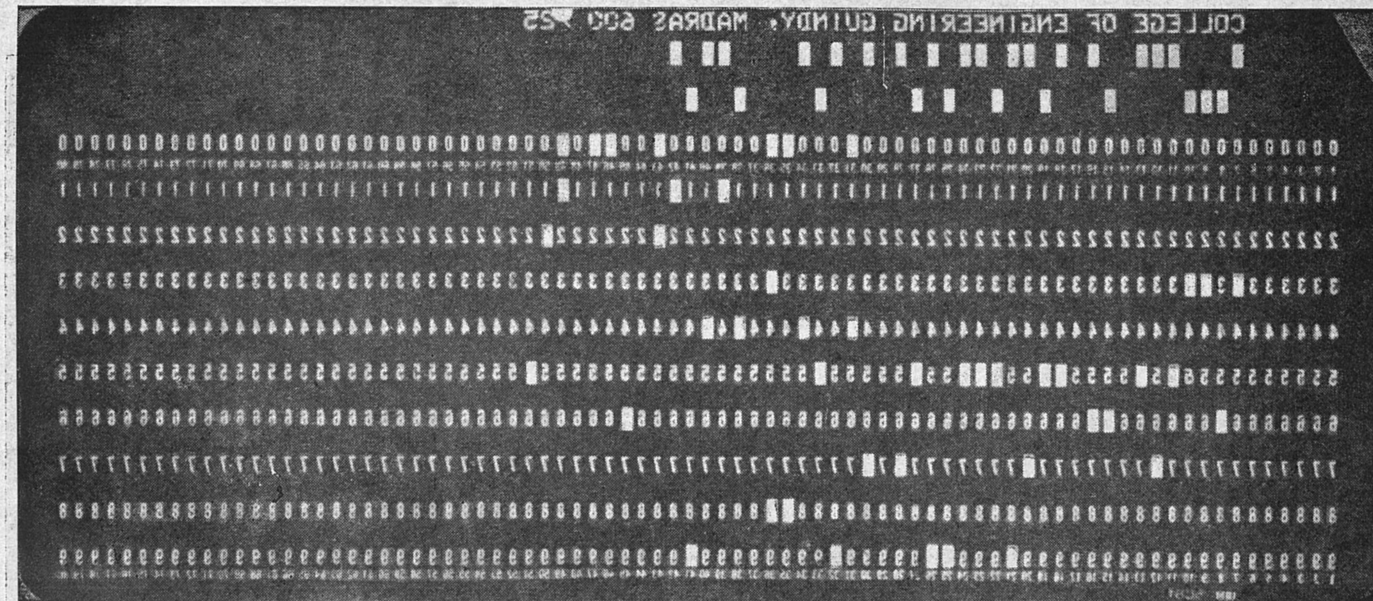
வரும் + குறி துளைக்கப்பட்ட இடங்களையும் தாண்டிச் செல் என்பதாகும். மேற்கண்ட புரோக்கிராம் கார்டில், இம்முறைப் படி *1* முதல் 9 முடிய உள்ள கார்டு இடங்கள் தாண்டிச் செல்லப்படுகின்றன. இவ்வாறு தாண்டிச் செல்லும் போது துளைமேடையில் ஒரு கார்டு இருப்பின் அதுவும் புரோக்கிராம் கார்டு நகரும்போது தானாகவே நகருகிறது; புரோக்கிராம் கார்டு *10*-ம் இடத்தில் வந்து நிற்கும் போது, துளைமேடைக் கார்டும் 10-ம் இடத்தில் வந்து காத்துக் கொண்டிருக்கும். புரோக்கிராம் கார்டில் பத்தாவது இடத்தில் துளை ஏதும் இல்லை. இதன் பொருள் இந்த இடத்திலும் இதை அடுத்துவரும் + குறி துளையிடப்பட்டுள்ள இடங்களிலும் எண்-நிலையில் (Numeric Modes) துளையிடப்பட வேண்டும் என்பதாம். இதன்படி இப் போது துளைமேடையில் உள்ள கார்டில் *10*-ம் இடத்திலிருந்து 19-ம் இடம் முடிய உள்ள இடங்களில் எண்களைத் துளை செய்யலாம். (இப்போது நாம் NUM பித்தானை அழுத்த வேண்டியதில்லை.) இதன் பின்னர் *20*-லிருந்து *24* இடங்கள் முடிய உள்ள இடங்களைப் புரோக்கிராம் உருளை தாண்டிச் சென்று (*20*-ம் இடத்தில் '—' குறியும் *21* முதல் *24* முடிய + குறி களும் இருப்பதை நோக்குக.) 25-ம் இடத்தில் வந்து நிற்கிறது. இப்போது துளைமேடைக் கார்டும் 25-ம் இடத்தில் துளைக்கப் பட காத்திருக்கிறது. புரோக்கிராம் கார்டில் 25-வது இடத்தில் '*1*' என்ற எண்ணும் 26 முதல் *44* முடிய உள்ள இடங்களில் A என்ற எழுத்தும் துளைக்கப் பட்டிருக்கின்றன. (A-ன் துளைக் குறி, + க்கான துளைக்குறியும் (*12*-ம் வரிசைத் துளை) *1*-க்கான துளைக் குறியும் (*1*-வரிசைத் துளை) சேர்ந்தது என் பதை நோக்குக. புரோக்கிராம் கார்டின் இத்துளைகளின் பொருள் 25-ம் இடத்தில் ஆரம்பித்து அடுத்துள்ள 'A' துளைக் கப்பட்டுள்ள இடங்கள் முடிய எழுத்து-நிலை (Alphabetic Mode) யில் துளை செய்யப்படும் என்பதாம். எனவே துளைமேடையில் உள்ள கார்டில், இப்போது A முதல் Z வரை உள்ள எழுத்துக் களையும் மற்ற எழுத்து நிலைக்குறிகளையும் பயன்படுத்தி பெயர் களைத் துளை செய்யலாம். இது முடிந்தவுடன் 45-ம் இடத்தி லிருந்து *80*-ஆவது இடம் முடிய, அதாவது கார்டில் மீதமுள்ள பகுதி, கடந்து செல்லப்பட்டுக் கார்டு முடிவடைகிறது. (45-முதல் *80*-ம் இடம் வரைக்காண, புரோக்கிராம் கார்டின் துளைக் குறிகளை நோக்குக.) இந்தக் கார்டு முடிவடைந்ததும், அடுத்த கார்டு எடுத்துக்கொள்ளப்பட்டு இதே முறையில் துளைக்கப் படுகிறது. அடுத்து வரும் அட்டவணையில் உருளைக் கார்டின் பலவித செயல்களையும் அவற்றிற்கான கார்டுக் குறியீடுகளும் கொடுக்கப் பட்டுள்ளன.

குறியீடு (Code)	செயல் (Function)	விளக்கம் (Description)
+ (கூட்டல்குறி)	இது வெறும் களக் குறி (Field Code) மட்டுமே	ஒவ்வொரு களத்திலும் தலையிடும் (High order positon) தவிர மற்ற இடங்களில் 12 பகுதி துளையிடப்பட வேண்டும்
— கழித்தல்குறி	கடந்துசெல்	ஒரு களத்தில் தலையிடத்தில் 11 பகுதியில் அதாவது — குறி துளையிட்டிருந்தால் அவ்விடத்திலிருந்து அடுத்துவரும் 12 பகுதித் துளையுள்ள இடங்களைக் (+ குறியுள்ள இடங்களை) கடந்து செல்லவேண்டும்.
O	பிரதிசெய் (Duplicate)	ஒரு களத்தில் தலையிடத்தில் O துளையிடப்பட்டிருந்தால் அந்த இடத்திலும் அடுத்துள்ள + குறி துளையிடப்பட்டுள்ள எல்லா இடங்களிலும் இடை மேடையில் உள்ள கார்டின் உரிய இடங்களில் உள்ள விவரங்களைப் பிரதியாகத் துளையிட வேண்டும்

குறியீடு	செயல்	விளக்கம்
1	எழுத்து நிலைக்கு மாற்றம்	ஒரு களத்தில் தலைக்குறி 1 ஆகவும் மீதி இடங்களில் A என்ற எழுத்தும் துளையிடப்பட்டிருப்பின் இக் களத்தில் எழுத்துக்கள் மட்டுமே துளையிடப்படும். (சாதாரணமாக உருளைக் கார்டின் கட்டுப்பாட்டில் துளைப்பொறி இயங்கும்போது அது 'எண்-நிலையில்' தான் இயங்கும்.) குறியிட்ட களத்திற்கு எழுத்து நிலைக்கு மாற்ற இக் குறியீடு தேவை
வெற்றிடம் (Blank)	எண் நிலைக்கு மாற்றம்	ஒரு களத்தில் தலையிடம் வெறுமையாகவும், மீதி யிடங்களில் + குறிகளும் காணப்படும் இப் பகுதியில் எண் நிலையில் துளைகளிடப்படும் என்று பொருள்.

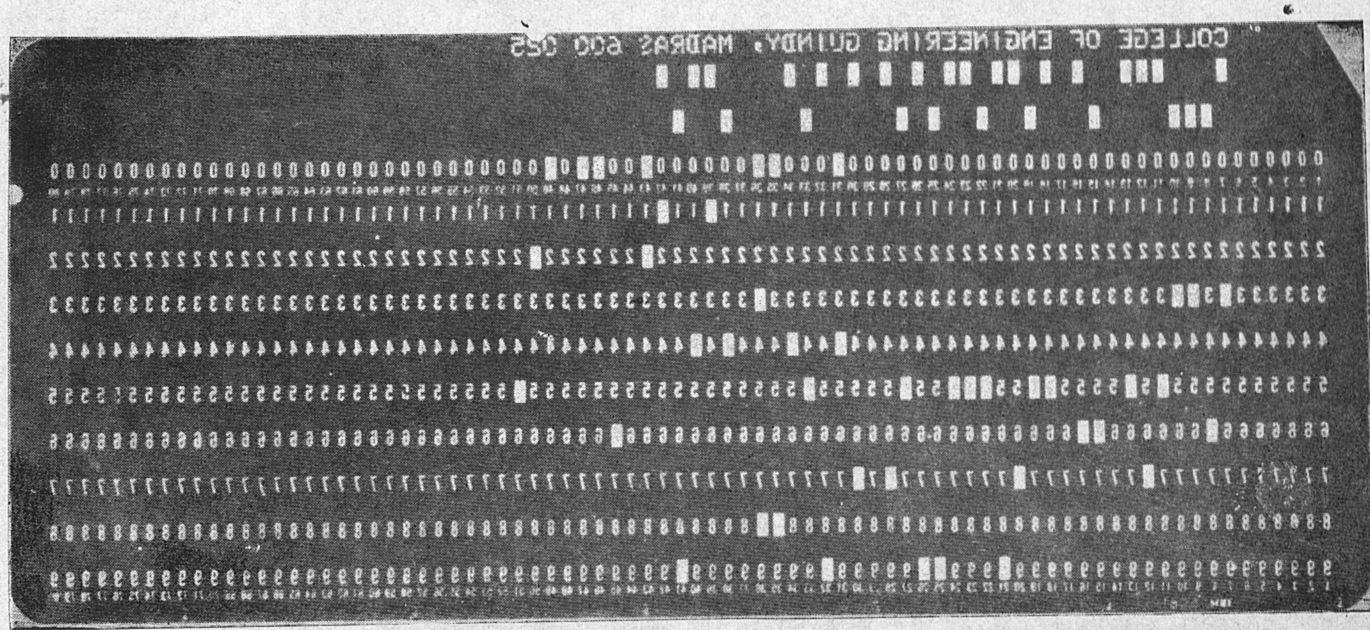
சரிபார்த்தி (Verifier)

துளைப் பொறிகளில் துளைக்கப்பட்ட கார்டுகளை சரிபார்த்தல் மிகவும் அவசியம். எவ்வாறு சாதாரணமாக டைப் செய்யும் போது தவறுகள் நேரக்கூடுமோ அவ்வாறே துளைப்பொறி மூலம் கார்டுகளைத் துளைக்கும் போதும் தவறுகள் ஏற்படலாம். இத் தவறுகளைக் கண்டுபிடிக்க 'கார்டு—சரிபார்த்தி'ப் பொறிகள் (Card Verifying Machines) உள்ளன. இப்பொறி பார்வைக்கு ஒரு துளைப் பொறியைப் போலவே இருக்கும். ஆனால் இதில் துளைப்பிடம் ஏதும் இல்லை. ஏற்கெனவே துளைக்கப்பட்ட கார்டைத் 'துளைமேடை'யில் ஏற்ற வேண்டும். கார்டில் முன்னரே துளைக்கப்பட்டுள்ள விவரங்களை மீண்டும் சரிபார்த்தி யின் பித்தான் முகத்தை இயக்கிப் பதிய வேண்டும். கார்டின் ஒரு இடத்தில் முன்னரே துளையிடப்பட்ட விவரமும் இப்போது பதியப்பட்ட விவரமும் ஒன்றேயானால் கார்டு அடுத்த இடத் திற்கு நகரும். வேறுபாடு இருப்பின், ஒரு செந்நிற விளக் கொன்று ஏற்றப்பட்டு கார்டு நகருவது நின்றுவிடும். (இந்த வேறுபாடு இரண்டு விதங்களில் ஏற்பட்டு இருக்கக் கூடும். ஒன்று துளைக்கப்பட்ட விவரம் சரியாக இருந்து பின்னர் சரி பார்த்தியில் பதிக்கப்பட்ட விவரம் தவருக இருக்கலாம். அதா வது சரிபார்த்தியை இயக்குபவர் கவனக்குறைவாக தவருன விவரத்தைப் பதியலாம். மற்ருென்று துளைக்கப்பட்ட விவரம் தவருகவே இருக்கக் கூடும்.) இப்போது ஒரு முன்னிலை (Reset) பித்தான் ஒன்றை அழுத்திஞல் எரிந்த செவ்விளக்கு அணைய, கார்டு தவறு நேர்ந்த இடத்திற்கே மீண்டும் சென்று நிற்கிறது. நாம் மீண்டும் கார்டில் **அதே இடத்தில்** உரிய விவரத்தைப் பதியவேண்டும். இப்போது பதியப்பட்ட விவரத்திற்கும் துளைக் கப்பட்டதற்கும் வேறுபாடு இல்லையாயின் கார்டு நகருகிறது; வேறுபாடு இருப்பின் மீண்டும் செந்நிற விளக்கு எரியும். இப் போது முந்நிலைப் (Reset) பித்தானை அழுத்திஞல் கார்டு தவறு நேர்ந்த இடத்திற்கு அடுத்த இடத்திற்கு நகருகிறது. ஆனல் நகருவதற்கு முன்னர் கார்டின்மேல் ஓரத்தில் தவறு உள்ள இடத்தில் ஓர் அரைவட்ட வெட்டு இடப்படுகிறது. இவ்வெட்டு, தவறு இருக்கும் இடத்தை நாம் அறிந்து கொள்ள உதவுகிறது. ஒரு கார்டில் ஒன்றுக்கு மேற்பட்ட இடங்களில் தவறு ஏற்படின் ஒவ்வொரு தவருன இடத்திற்கும் எதிரே கார்டின் விளிம்பில் அரைவட்ட வெட்டு இடப்படும். ஒரு கார்டில் தவறு ஏதும் இல்லையாயின் வலதுபுற ஓரத்தில் மேற்பகுதியில் ஒரு அரை வட்ட வெட்டு இடப்படும்.



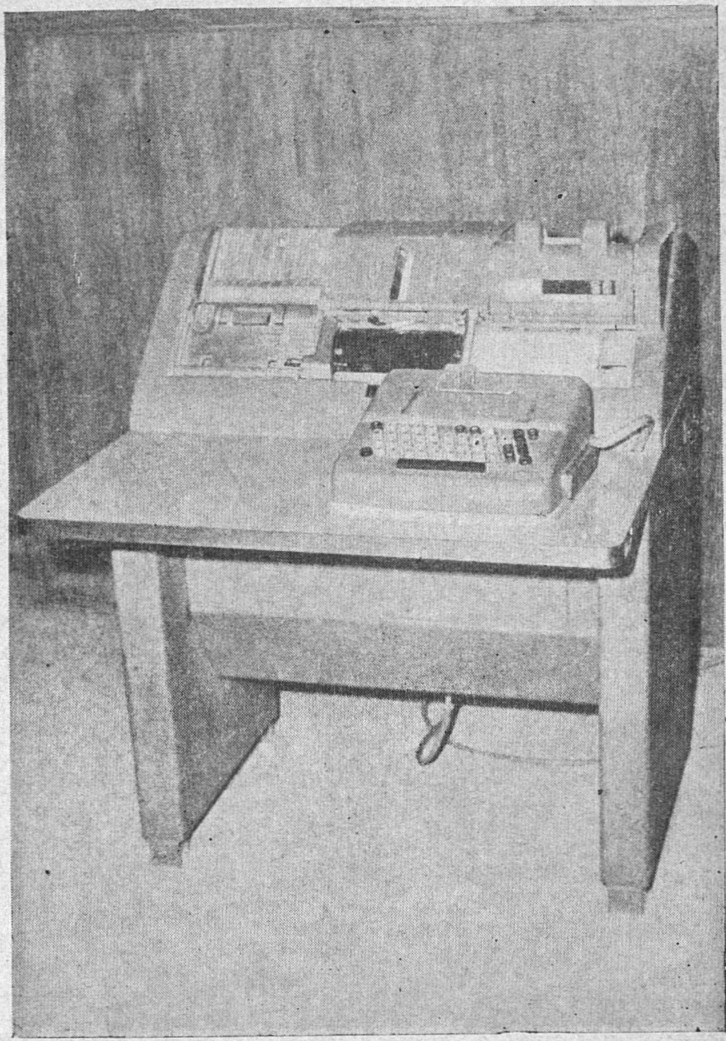
படம் 4.6a

தவருகத் துளைக்கப்பட்ட கார்டு-சரிபார்த்தியில் பயன்படுத்தப்பட்ட பிறகு
(மேற்புற ஓரத்தில் உள்ள அரைவட்ட வெட்டைக் கவனிக்கவும்)



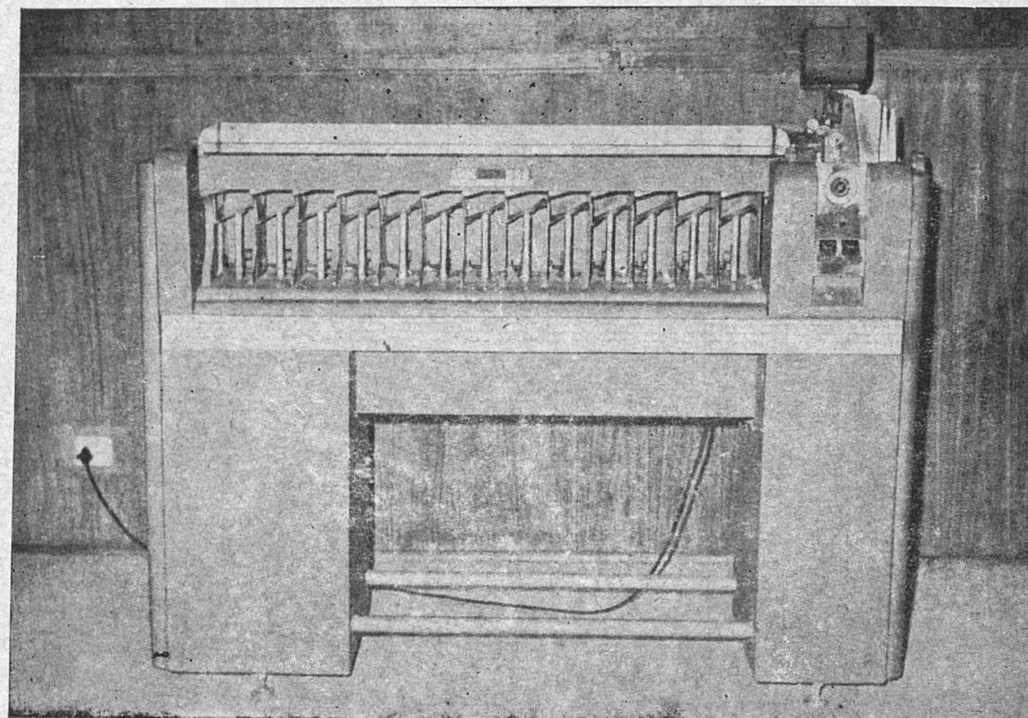
படம் 4.6b

சரியாகத் துளைக்கப்பட்ட கார்டு-சரிபார்த்தியில் பயன்படுத்தப்பட்ட பின்னர்
(வலப்புற ஓரத்தில் உள்ள அரைவட்ட வெட்டைக் கவனிக்கவும்)



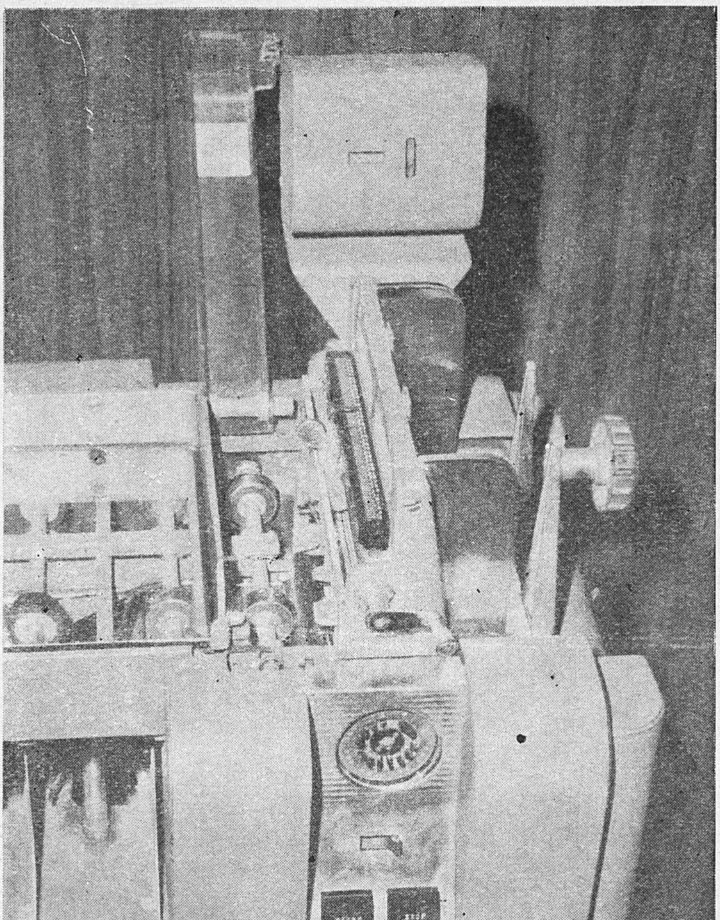
படம் 4.7

சரிபார்த்தி (Card Verifier)



படம் 4.8

சார்ட்டர் (Sorter—முறைப்படுத்தி)



படம் 4.9

முறைப்படுத்தியின் படிப்பிடம் (Reading Station)

வகைப்படுத்தி (Sorter)

அநேக ஆய்வுகளில் துளைக்கார்டுகளை அவற்றில் உள்ள விவரங்களுக்கு ஏற்றவாறு முறைப்படுத்தல் (Sorting) அவசிய மாகும். சிலவகைக் கணிப்புகளில் கார்டுகளை அகர வரிசையில் (Alphabetic Order) அடுக்கவேண்டியிருக்கும். இன்னும் சில வற்றில் குறிப்பிட்ட எண்களின் ஏறும் அல்லது இறங்கும் வரிசை யில் கார்டுகளை அடுக்க வேண்டியிருக்கலாம். கையாளப்படும் கார்டுகளின் எண்ணிக்கை ஆயிரக்கணக்கில் இருக்கும்போது அவற்றைக் கையால் பிரித்து அடுக்குவதென்பது இயலாத காரியம். இவற்றை வகைப்படுத்துவதற்கென்று வகைப்படுத்தி கள் (Sorters) உள்ளன.

வகைப்படுத்தியின் முக்கிய பாகங்கள் (1) கார்டு இடுமிடம், (2) படிப்பிடம், (3) சேகரிப்பறைகள், (4) தேர்வுப் பித்தான் கள் ஆகியவையாகும். வகைப்படுத்தி எனப்படும் வகைப் படுத்தியின் படிப்பிடம் (படம் 4.9) ஒரு நீண்ட பித்தளை உருளையும் அதன்மீது தொட்டுக் கொண்டிருக்கும் ஒரு ஒற்றை பிரஷ்ஷையும் கொண்டது. இந்த ஒற்றை பிரஷ்ஷை பித்தளை உருளையின் மீது, சமதூரத்தில் அமைந்த 80 வெவ்வேறு இடங் களில் நகர்த்திப் பொருத்தலாம். இந்த 80 இடங்களும் ஒரு கார்டின் துணையேற்கும் 80 இடங்களுடன் நேரடித் தொடர்பு கொண்டவை. முறைப் பொறியில் உள்ள சேகரிப்பறைகள் 13 ஆகும். அவை முறையே இடமிருந்து வலமாக 9, 8, 7, 6, 5, 4, 3, 2, 1, 0, 11, 12, R என்ற பெயர் கொண்டவை.

இப்போது வகைப்படுத்தி எவ்வாறு செயல்படுகிறது என்று பார்ப்போம். வகைப்படுத்தியின் பிரஷ்ஷை, படிப்பிடத்தில் 7-ஆம் இடத்தில் இருப்பதாக வைத்துதகொள்ளுவோம். இப் போது பொறியின் கார்டு இடுமிடத்தில் நன்ருக அடுக்கிய ஒரு கட்டுக் கார்டை, கட்டின் முகம் அடியிலும் 9 ஓரம் பிரஷ் பக்கமும் இருக்குமாறு வைத்துப் பொறியை இயக்குவோம். இப் போது கார்டுகள் ஒவ்வொன்ருக படிப்பிடத்தின் வழியே செல்லு கின்றன. அவ்வாறு செல்லும்போது 7-ம் இடத்தில் உள்ள துளைகள் மட்டுமே படிக்கப்படுகின்றன. (பிரஷ்ஷை 7-ம் இடத்தில் பொருத்தி யிருக்கிரேம் என்பதை நினைவு கூர்க) ஏதாவது ஒரு கார்டில் 7-வது இடத்தில் துளை ஏதுமில்லையாயின் அக்கார்டு 'R' (Reject) என்று குறிக்கப்பட்டுள்ள அறையில் வந்துவிழும். முறைபடுத்த எடுத்துக்கொள்ளப்பட்ட ஒரு கார்டில் 7-வது இடத்தில் எந்த இடத்தில் துளை இடப்பட்டிருக்கிறதோ

அந்த எண்ணைக்கொண்ட சேகரிப்பறைக்கு அது எடுத்துச் செல்லப்பட்டு அங்கே சேகரிக்கப்படும். எடுத்க்காட்டாக ஒரு கார்டில் 7-வது இடத்தில் 3 என்ற எண் துளையிடப்பட்டிருக் கிறதெனக் கொள்ளுவோம். இந்தக் கார்டு '3' என்று குறிக் கப்பட்ட சேகரிப்பறையில் சென்று விழும்.

வகைப்படுத்தியைக்கொண்டு, கார்டு கட்டுகளை, எண்களின் ஏறும் அல்லது இறங்கும் வரிசையில் முறைப்படுத்தலாம் என்று கூறினோம். இது எவ்வாறு செய்யப்படுகிறது என்பதைக் காண் போம். எடுத்துக்காட்டாக ஒரு தொழிற்சாலையின் தொழிலாளி களின் விவரங்கள் கார்டுகளில் துளைக்கப்பட்டிருக்கின்றன எனக் கொள்ளுவோம். ஒவ்வொரு கார்டிலும், ஒரு தொழிலாளியின் விவரங்கள் கொடுக்கப்பட்டுள்ளதாயும், ஒவ்வொரு தொழிலா ளிக்கும், ஒரு மூவிலக்க 'தொழிலாளி எண்' கொடுக்கப்பட் டுள்ளது என்றும், ஒவ்வொரு கார்டிலும் இந்த மூவிலக்க எண், முதல் மூன்று இடங்களில் துளைக்கப்பட்டிருப்பதாகவும் கொள்ளு வோம். நமக்குக் கொடுக்கப்பட்டிருக்கும் கட்டில் உள்ள கார் டுகள் இப்போது இருக்கும் நிலையில் 'தொழிலாளி எண்'களின் ஏறும் வரிசையில் இல்லை என்று வைத்துக்கொள்ளுவோம். சீர்ப் படுத்தி இக் கார்டுக் கட்டை 'தொழிலாளி எண்களின்' ஏறும் வரிசையில் அமைக்கவேண்டும். அதைச் செய்வது எவ்வாறு?

முறைப்பொறியில் உள்ள பிரஷ்ஷை முதலில் மூன்றும் இடத் தில் பொருத்தி நமது கார்டுகளை பிரிக்கவேண்டும். (இந்த மூன்றும் இடம் நமது தொழிலாளி எண்ணின் ஒன்றும் நிலை என்பதை நினைவில் கொள்க) சேகரிப்பறைகளில் வந்து விழுந்த கார்டுகளை முதலில் 0-அறைக் கார்டுகளையும், அதற்கு பின்புறத் தில் 1-அறையில் விழுந்த கார்டுகளையும், அதற்கும் பின்புறத் தில் 2-அறையில் விழுந்த கார்டுகளையும், இவ்வாறே ஒன்றன் பின் ஒன்றாக சேகரிப்பு அறையில் விழுந்த கார்டுகளை அடுக்கி, இறுதியில் 9-அறைக் கார்டுகள் இருக்குமாறு சேகரித்துக்கொள்ள வேண்டும். (இப்போது நமது மூன்று இலக்க எண்களின் ஒன்றும் நிலை ஏறும்- வரிசையில் இருக்குமாறு அடுக்கியுள்ளோம்.) அடுத்தபடியாக, முறைப் பொறியின் பிரஷ்ஷை 2-ம் இடத் திற்குத் (அதாவது தொழிலாளி எண்ணின் பத்தாம் நிலையில்) தள்ளி வைத்து புதிதாக அடுக்கப்பட்ட கார்டு கட்டை முறைப் பொறியிட்டு மீண்டும் பிரிக்கவேண்டும். முன் போலவே இப் போது சேகரிப்பு அறைகளில் விழுந்துள்ள கார்டுகள் 0- அறைக் கார்டுகள் மேலிருக்குமாறும் அதற்கடியில் 1- அறைக்கார்டுகள், அதன் பின்னர் 2-அறைக்கார்டுகள், இவ்வாறே இறுதியில் 9-

அறைக்கார்டுகள் இருக்குமாறும் அடுக்க வேண்டும். இப்போது 10-ம் நிலை அடுக்கலை முடித்துள்ளோம். இதைப் போலவே முறைப்பொறியின் பிரஷ்ஷை முதலிடத்திருந்துத் தள்ளி வைத்து, கார்டுகளை மீண்டும் 100-ம் நிலைக்கு முறைப்படுத்த வேண்டும். இவ்வாறு மும்முறை கார்டுகளை வகைப்படுத்தி ஓட்டி அடுக்கிய பின்னர் நமது கார்டுக்கட்டு 'தொழிலாளி எண்களின்' ஏறும் வரிசையில் அமைந்திருக்கும். இதே போல் நான்கு ஐந்து முதலிய நிலை எண்களை ஏறும் வரிசையிலும் நாம் அடுக்கலாம்; எண்களில் எத்தனை நிலைகள் உள்ளனவோ அத்தனை தடவை கார்டுக்கட்டை முறைப்பொறியில் ஓட்ட வேண்டியிருக்கும். எடுத்துக்காட்டாக ஒரு கார்டுக்கட்டில் உள்ள கார்டு ஒவ்வொன்றி லும் 10, 11, 12, 13, 14 ஆகிய இடங்களில் இருக்குமாறு ஐந்து இலக்க எண்கள் துளையிடப்பட்டிருப்பின் இக்கட்டை இவ்வெண் களில் ஏறும் வரிசையில் அடுக்க முதலில், வகைப்படுத்தியின் பிரஷ்ஷை 14-ம் இடத்தில் (ஒன்றும் நிலை) பொருத்தி கார்டுகளை வகைப்படுத்தியில் ஓட்ட வேண்டும். ஓட்டிய பின்னர் முன் கூறியவாறு கார்டுகளை அடுக்கி, அதன் பின்னர், பிரஷ்ஷை முறையே 13, 12, 11, 10 இடங்களில், (பத்து, நூறு, ஆயிரம். பத்தாயிரம் நிலைகளில்) ஒன்றன் பின் ஒன்றுகத் தள்ளி வைத்து, கார்டுகளை மீண்டும் நான்கு முறை ஓட்டிப் பிரித்து அடுக்க வேண்டும்.

மேற்சொன்ன முறை எவ்வாறு செயல்படுகிறது என்பதை ஒரு சிறு எடுத்துக்காட்டு மூலம் காண்போம். ஒரு கட்டில் 10 கார்டுகள் இருப்பதாகவும், ஒவ்வொரு கார்டிலும் ஒரு மூவிலக்க எண் துளைக்கப் பட்டிருப்பதாகவும் கொள்ளுவோம். இந்தக் கட்டில் உள்ள கார்டுகளின் வரிசைப்படி அவற்றில் துளைக்கப் பட்டுள்ள இலக்கங்கள் கீழுள்ள அட்டவணையில் வரிசை 0-ல் காட்டப்பட்டுள்ளன. இதை ஏறும் வரிசையில் அடுக்கவேண்டு மென்று வைத்துக்கொள்ளுவோம்.

வகைப்படுத்துமுன் உள்ள கார்டுகளின் வரிசையை 0 வரிசை என்போம். 0-வரிசை எண்களின் 1-ம் நிலையைமட்டும் கவனத்தில் கொண்டு 1-ம் நிலையின் ஏறும் வரிசையில் (எண் கள் கொடுக்கப்பட்டுள்ள ஒழுங்கில்) (Order) இம் மூவிலக்க எண்களைத் திருப்பி அடுக்கினுல், இவை 1-ம் வரிசையில் உள்ள ஒழுங்கில் (Order) காணப்படும். வரிசை 1-ல் உள்ள எண்களை அவை இருக்கும் ஒழுங்கில், 10-ம் நிலை ஏறும் வரிசையில் அடுக்க, வரிசை 2-கிடைக்கிறது. மீண்டும் வரிசை 2-ஐ 100-ம் நிலை

ஏறும் வரிசையில் அடுக்கினால் வரிசை-3 கிடைக்கிறது. வரிசை 0-வின் ஏறும் வரிசையில் அடுக்கப்பட்ட மறுவடிவமே வரிசை-3 என்பதை நோக்குக:

வரிசை 0	வரிசை 1	வரிசை 2	வரிசை 3
953	590	500	003
093	471	003	093
536	953	107	107
471	093	113	113
500	113	216	216
113	003	536	471
107	536	536	500
536	536	953	536
216	216	471	536
003	107	093	953

வகைப்படுத்தியைக் கொண்டு எண்களின் ஏறும் / இறங்கும் வரிசையில் கார்டுகளை அடுக்குவதுமட்டுமின்றி எழுத்துக்களின் அகர வரிசையிலும் அடுக்க இயலும்.

வகைப்படுத்தியின் தேர்வுப் பித்தான்களில் 'அகரப்பித்தான்' (Alphabetic Sorting Switch) ஒன்றும் உள்ளது. இதையும் மற்ற பித்தான்களையும் முன்னும் பின்னுமாக தள்ளி வைக்க இயலும். கார்டுகளை எண்களின் வரிசையில் பிரிக்கும் போது, தேர்வுப் பித்தான்கள் யாவும் பின்புறமாகத் தள்ளப்பட்டிருக்க வேண்டும். அகர வரிசையில் பிரிக்க வேண்டுமானால் அகரப் பித்தானை மட்டும் முன்தள்ளி வைத்து மற்றவற்றைப் பின்தள்ளி வைக்க

வேண்டும். எடுத்துக்காட்டாக ஒரு கட்டுக் கார்டில் 10-ம் இடத்திலிருந்து பெயர்கள் துளைக்கப்பட்டிருப்பதாகக் கொள்ளு வோம். வகைப்படுத்தியின் பிரஷ்ஷை 10-ம் இடத்தில் இருத்தி, தேர்வுப்பித்தான்களை அகரவரிசைப் பிரிப்புக்கு ஏற்றவாறு அமைத்துக் கார்டுகளை வகைப்படுத்தியிலிட்டு ஓட்டுவோம். தேர்வுப் பித்தான்கள் அகர நிலையில் இருக்கும் போது கார்டில் 1, 2, 3......9 ஆகிய துளைகள் "புறக்கணிக்கப்படும்." 0, 11, 12 ஆகிய மூன்று இடங்களில் உள்ள துளைகளே 'படிக்கப்படும்'. நமது கார்டுகட்டில் 10-ஆம் இடத்திலிருந்து பெயர்கள் துளைக்கப் பட்டிருப்பதால், அவ்விடத்தில் 0, 11, 12, ஆகிய மூன்றில் ஏதாவது ஒன்றில் அவசியம் துளையிருந்தாக வேண்டும். ஆகவே நமது கார்டுகட்டு 0, 11, 12, ஆகிய மூன்று சேகரிப்பு அறை களில் சேகரிக்கப்படுகிறது. இவ்வாறு சேகரிக்கப்பட்ட கார்டுகளை நாம் தனித்தனிக் கற்றைகளாக வைத்துக் கொள்ள வேண்டும். அடுத்து முறைப்பொறியின் பிரஷ்ஷை இடம் மாற்றுமல் அகர பித்தானை பின்புறம் தள்ளிவைத்துக் கொள்ள வேண்டும். பின்னர் 0- கற்றை, 11-கற்றை, 12-கற்றை ஆகியவற்றைத் தனித்தனியாக வகைப்படுத்தியிலிட்டு, எண்ணின் ஏறும் வரிசை யில் பிரித்துக் கொள்ள வேண்டும். பிரித்துக் கொண்டவுடன் சேகரிப்பு அறைகளில் இருக்கும் கார்டுகளில் 1-துளைக்கார்டுகளை மேற்புறமும் 2-துளைக்கார்டுகளை அதற்கடுத்தும், 3-துளைக் கார்டுகளை அதற்குப் பின்னரும்......இவ்வாறே கடைசியாக 9-துளைக்கார்டுகளை எல்லாவற்றிற்கும் அடியிலும் இருக்குமாறு வரிசையாக சேகரித்து அடுக்கிக் கொள்ள வேண்டும். இவ்வாறு அடுக்கிய பின்னர், 12-கற்றைக் கார்டில் 10-ஆம் இட எழுத்துக் கள் A,B,C,D,E,F,G,H,I என்ற வரிசையிலும் 11-கற்றையில், 10-ஆம் இட எழுத்துக்கள் J,K,L,M,N,O,P,Q,R என்ற ஒழுங்கிலும் 0-கற்றையில் S,T,U,V,W,X,Y,Z என்ற முறை யிலும் இருக்கும். இம்மூன்று கற்றைகளையும் முதலில் 12-கற்றை யும் மேலே அதன் பின்னர் 11 கற்றையும், கடைசியில் 0-கற்றை யும் வருமாறு அடுக்க வேண்டும். இப்போது நமது கார்டுகற்றை 10ம் இட எழுத்துக்களைப் பொறுத்தவரை அகரவரிசையில் அடுக்கப்பட்டு இருக்கும்.

வகைப்படுத்தியை வேறொரு வகையான முறைப்படுத் தலுக்கும் பயன்படுத்தலாம். ஒரு கற்றைக்கார்டில் குறிப்பிட்ட இடத்தில் ஒரு குறிப்பிட்ட குறியுள்ள கார்டுகளை மட்டும், மற்றக் கார்டுகளின் ஒழுங்குமாறுமல் பிரித்து எடுக்க இயலும். எடுத்துக் காட்டாக ஒரு கற்றைக் கார்டில், 15-ம் இடத்தில் 2,9 ஆகிய இரு இடங்களில் துளையிடப்பட்ட கார்டுகள் மட்டும் பிரித்தெடுக்

கப்பட வேண்டும்; மற்றகார்டுகளின் ஒழுங்கும் மாறலாகாது. இதைச் செய்து முடிக்க, வகைப்படுத்தியின் பிரஷ்ஷை 15-ம் இடத்தில் பொருத்திய' பின்னர், தேர்வுப் பித்தான்களில் 2,9 ஆகிய இரண்டை மட்டும் பின்னுல் வைத்துக் கொண்டு, மற்றதை மூன்னுல் தள்ளி வைத்துக் கொள்ள வேண்டும். இப்போது கார்டுக் கற்றையை வகைப்படுத்திவிட்டு இயக்க, எந்தெந்த கார்டில் 15-ம் இடத்தில் 2 அல்லது 9 துளைக்கப்பட்டிருக்கிறதோ அந்தக் கார்டுகள் மட்டும் அவற்றிற்குரிய 2 அல்லது 9-வது எண் சேகரிப்பு அறையில் சென்றுவிழும். ஏனைய கார்டுகள் யாவுமே. R- அறையில் ஒழுங்குமாருமல் வந்து விழும்.

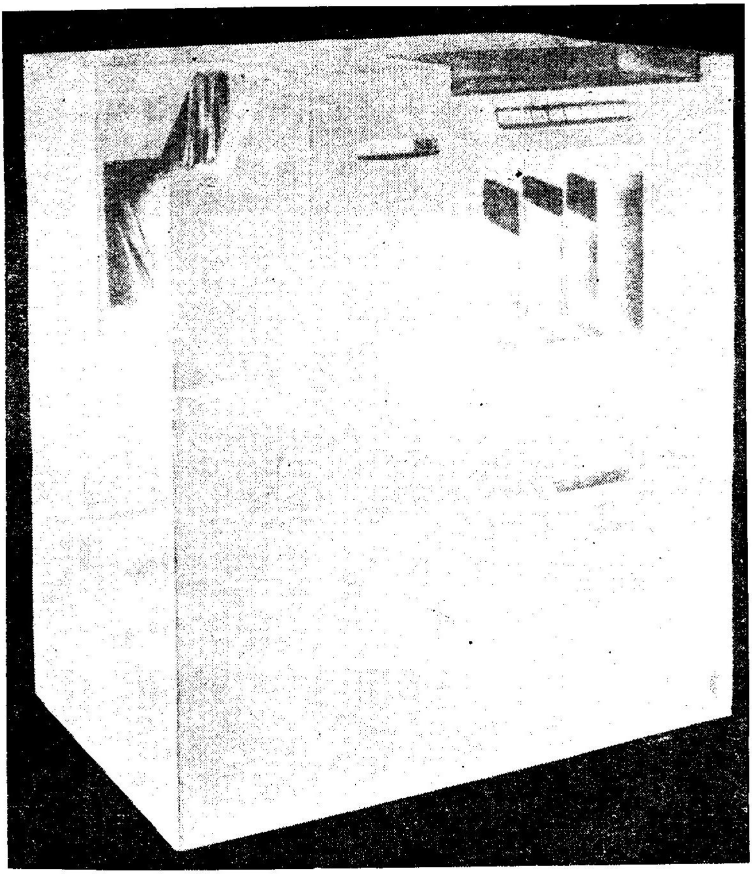
வேகத்தை பொறுத்தவரை பல திறப்பட்ட வகைப் படுத்திகள் உள்ளன. ஐ. பி. எம். 80 வகைப்படுத்தியில் சராசரி ஒரு நிமிடத்திற்கு 650 கார்டுகளும், ஐ. பி. எம். 83-ல் ஒரு நிமிடத் திற்கு 1200 கார்டுகளும் முறைப்படுத்தப்படுகின்றன. நாம் முன் பார்த்த விவரங்கள் யாவும் ஐ. பி. எம். 80 வகைப் படுத்தியைப் பற்றியதே. ஐ. பி. எம். 83-ல் அதிகப்படி வேகத்தைவிட வேறு சில சௌகரியங்களும் உண்டு.

" இணைப்பி " (Collater)

யூனிட் ரெக்கார்டு மெஷின்களில், கொலேட்டர் எனப்படும் இணைப்பி முக்கியமான ஒன்றும். கொலேட்டரின் பணிகள் முக்கியமாக நான்கு ஆகும். அவை (i) வரிசைச் சோதிப்பு (Sequeuce checking) (ii) ஒன்றல் (Merging) (iii) இணைத்தல் (Matching) (iv) தேர்வு (Selecting) ஆகியவையாம். இவற்றைப் பற்றிச் சற்று பின்னர் பார்ப்போம்.

கொலேட்டரில் இரண்டு தனித்தனியான கார்டு இடுமிடங் கள் உள்ளன. ஒன்றைப் பிரைமரி (Primary) என்றும் மற்றதை செகண்டரி (Secondary) என்றும் கூறுவர். மற்ற யூனிட் ரெக் கார்டு மெஷின்களில் உள்ளதைப் போலவே இதிலும் கார்டு சேகரிப்பு - இடங்கள் உள்ளன. இத்தகைய சேகரிப்பு அறைகள் 4 இருக்கின்றன.

இந்த அறைகளுக்கு வலமிருந்து இடமாக 1, 2, 3, 4 என்று எண் கொடுப்போம். இந்த நான்கு அறைகளுள் 4, 3, 2 எண் களைக்கொண்ட அறைகள் செகண்டரியிலிருந்து வரும் கார்டு களுக்கும் 1, 2 எண்களைக்கொண்ட இரு அறைகளும் பிரைமரி யிலிருந்து வரும் கார்டுகள் விழுவதற்கும் ஆனவை. இவ்வ மைப்பின்படி பிரைமரி, செகண்டரி இவை யிரண்டுக்கும் பொது



படம் 4.10
கொலேட்டர் அல்லது இணைப்பி

வானது அறை எண் 3 ஒன்றே. செகண்டரி கார்டு அறைக்கும், அதற்குறிய சேகரிப்பறைகளுக்கும் இடையே உள்ள வழியில் ஒரு படிப்பிடம் (செகண்டரி படிப்பிடம்) உள்ளது. இதற்கு மாருகப் பிரைமரி கார்டு அறைக்கும், அதற்குரிய சேகரிப்பறைகளுக்கும் இடையே இரண்டு படிப்பிடங்கள் உள்ளன. இவை இரண்டில் பிரைமரி அறைக்கு அருகில் உள்ளதற்கு வரிசை படிப்பிடம் என்றும் (Sequence Read) மற்றதற்கு பிரைமரி படிப்பிடம் என்றும் பெயர். பிரைமரி கார்டு அறையிலிருந்து சேகரிப்பறைகளுக்கு ஒரு கார்டு செல்லும்போது முதலில் அது வரிசைப்-படிப்பிடம் வழியாகவும், பின்னர் பிரைமரி படிப்பிடம் வழியாகவும் சென்று முடிவில் சேகரிப்பு அறையை அடைந்தாகவேண்டும். இப்போது கொலேட்டரின் நான்கு பணிகளைப் பற்றிப் பார்ப்போம்.

வரிசைச் சோதிப்பு (Sequence checking)

இச்சோதிப்பு, கொடுக்கப்பட்டுள்ள கட்டில் உள்ள கார்டுகள் ஏறும் வரிசையில் உள்ளனவா என்று சோதிப்பதேயாம். எடுத்துக்காட்டாக கற்றை கார்டில், ஒவ்வொன்றிலும் 5 முதல் 8 முடிய உள்ள இடங்களில் (Columns) நான்கு இலக்க எண்கள் துளைக்கப்பட்டிருக்கின்றன என்று கொள்ளுவோம். இந்த கற்றையை ஏற்கனவே வரிசைப்படுத்தியில் இட்டு இந்த 4 இலக்க எண்ணின் ஏறும் வரிசையில் அடுக்கி யிருக்கிறோம் என்றும் கொள்ளுவோம். இவ்வாறு அடுக்கப்பட்ட கார்டுகள் உண்மையிலேயே ஏறும் வரிசையில் இருக்கின்றனவா என்று நாம் இப்போது சோதிக்க வேண்டும். இந்தக் கார்டு கற்றையை பிரைமரி அறையில் வைத்து இயக்க, கார்டுகள் ஒன்றன் பின் ஒன்ருக வரிசை-படிப்பிடம், பிரைமரி-படிப்பிடம் ஆகியவற்றை கடந்து சேகரிப்பு அறை 2-ல் சென்று சேறுகின்றன. கட்டின் முதல் கார்டு பிரைமரி படிப்பிடத்தில் இருக்கும்போது, இரண்டாவது கார்டு வரிசை–படிப்பிடத்தில் இருக்கும். இவ்விரண்டு படிப்பிடங்களும் இரண்டு கார்டுகளிலும் 5-8 இடங்களில் உள்ள 4 இலக்க எண்களைப் படித்து ஒப்பிடுகின்றன. இந்த இரண்டு எண்களில் வரிசை-படிப்பிடக் கார்டின் எண், பிரைமரி படிப்பிடத்தில் உள்ள கார்டின் எண்ணை விடப் பெரியதாகவோ அல்லது அதற்கு சமமானதாகவோ இருப்பின், முதல் கார்டு சேகரிப்பறை 2-க்குள் சென்று விழும். வரிசைப்-படிப்பிடத்தில் இருந்த கார்டு இப்போது பிரைமரி - படிப்பிடத்தை வந்தடைய மூன்ருவது கார்டு ஒன்று வரிசை- படிப்பிடத்தை வந்து சேரும். இப்போதுமீண்டும் பிரைமரி, வரிசை ஆகிய படிப் பிடங்களில் இப்போதுள்ள கார்டுகளில் 5-8 இடங்களில் துளைக்கப்பட்டுள்ள

எண்கள் படிக்கப்பட்டு ஒப்பு நோக்கப்படுகின்றன. வரிசை-படிப்பிடக் கார்டில் காணும் எண், பிரைமரிப் படிப்பிடக் கார்டின் எண்ணை விடக் குறைவாக இருப்பின் அது ஒழுங்கு தவறிய முறையாம். இந்நிலை ஏற்பட்ட உடனே கொலேட்டரின் இயக்கம் நின்றுவிடும். அவ்விடத்தில் கார்டுகளின் வரிசையை நாமாக சரிப்படுத்தி மீண்டும் பொறியை இயக்கலாம்.

ஒன்றல் :—(Merging)

சில சமயங்களில் ஒரு குறிப்பிட்ட விவரத்தின் அடிப்படையில் வரிசையாக அடுக்கப்பட்ட இரு கட்டுக் கார்டுகளை ஒன்றை மற்றொன்றில் இடைச் செருக வேண்டியிருக்கும். இதில் ஒரு கட்டை, பிரைமரி கார்டு அறையிலும் மற்றதை செகண்டரி கார்டு அறையிலும் வைத்து கொலேட்டரை இயக்க வேண்டும். இந்த ஒன்றல் எவ்வாறு நடக்கிறது என்பதை அட்டவணை

ஒன்றலுக்கு முன்		ஒன்றலுக்குப் பின்
கார்டுகள் வரிசை		சேகரிப்பு
பிரைமரி	செகண்டரி	அறை 2
11	10	11
9	8	10
6	7	9
2	5	8
1	4	7
	3	6
		5
		4
		3
		2
		1

அட்டவணை 4.2

4.2 காட்டுகிறது. மேலுள்ள அட்டவணையில் 'பிரைமரி' என்பதன் அடியில் காணும் எண்கள் ஒவ்வொன்றும் ஒரு கார்டில் துளைக்கப்பட்டு அட்டவணையில் காணும் வரிசையில் (பிரைமரி கார்டு அறையில்) அடுக்கப்பட்டு உள்ளன; அவ்வாறே செகண்டரி

அறையின் கார்டுகளும் கடைசியில் உள்ள 'சேகரிப்பு அறை 2' என்பதன் கீழ்காணும் விவரம், பிரைமரி கார்டுகளும் செகண்டரி கார்டுகளும் ஒன்றிய பிறகு உள்ள வரிசையைக் காட்டுகிறது.

இணைத்தல் (Matching)

இணைத்தலும், ஒன்றலைப் போன்றதே. இந்த முறையில். வரிசையில் அடுக்கப்பட்ட இரண்டு கட்டுக் கார்டுகளுள் இணையான கார்டுகள் மட்டும் தனித்தனியாகச் சேகரிக்கப்படுகின்றன. அட்டவணை 4-3 ஐக் காண்க

இணைத்தலுக்கு முன்		இணைத்தலுக்குப் பின்			
பிரைமரிக் கார்டு அறை	செகண்டரி கார்டு அறை	சேகரிப்பு அறைகள்			
		1	2	3	4
4	5	4			5
3	3		3	3	
2	2		2	2	
1	1		1	1	

அட்டவணை 4.3

தேர்ந் தொன்றல் (Select and Merge)

சில வகை கணிப்புகளில், தனித் தனியாக வரிசை முறையில் அடுக்கப்பட்ட இரண்டு கட்டு கார்டுகளிலிருந்து இணையான கார்டுகளைத் தேர்ந்து அவற்றை ஒன்ற வேண்டியிருக்கும். இதையும் கொலேட்டரைக் கொண்டு செய்ய இயலும். இதைச் செயல்படுத்தியதும் ஏற்படும் விளைவை அடுத்துள்ள அட்டவணை காட்டுகிறது.

தேர்ந்திணைத்தல் :— (Select and Maching)

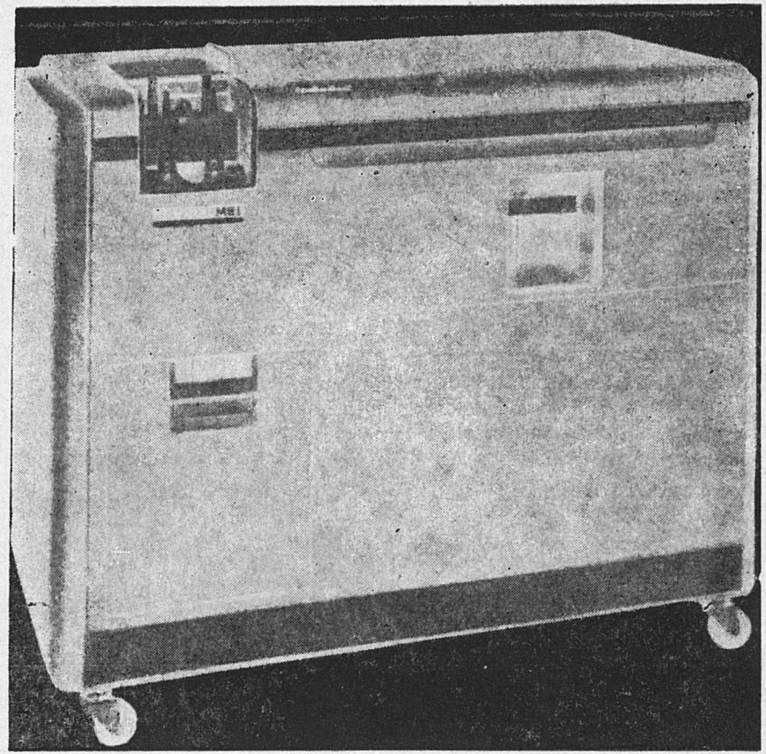
தேர்ந்தொன்றல் போன்றதே தேர்ந்திணைத்தலும் ஆகும். தேர்ந்தொன்றலில் இணையான கார்டுகள் ஒரே சேகரிப்பறையில் சேர்க்கப்படுகின்றன. ஆனால் தேர்ந்திணைத்தலில் இவை தனித் தனி சேகரிப்பறைகளில் சேமிக்கப்படுகின்றன. அடுத்துள்ள அட்டவணையைக் காண்க.

தேர்ந்தொன்றலுக்கு முன்னர்		தேர்ந்தொன்றலுக்கு பின்னர்			
பிரைமரி	செகண்டரி	4	3	2	1
6					
5	6				
4	4			6	
2	3			4	
1	1		6	1	5
			3	1	2

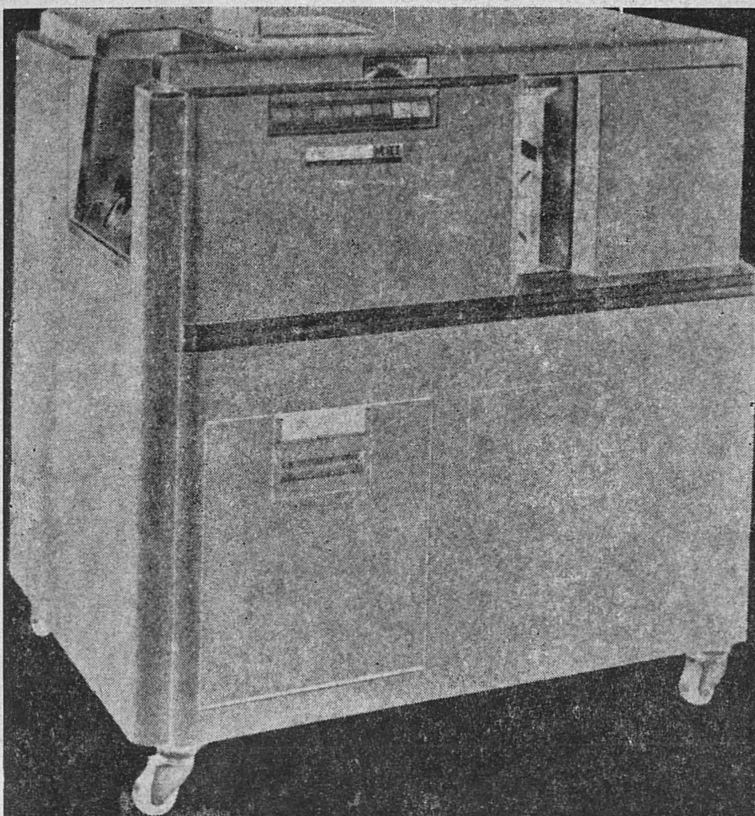
அட்டவணை 4 a

தேர்ந்திணைத்தலுக்கு முன்		தேர்ந்திணைத்தலுக்குப் பின்			
பிரைமரி	செகண்டரி	4	3	2	1
5	6				
4	5				
3	4		5	5	
2	2		4	4	
1	1		2	2	
		6	1	1	3

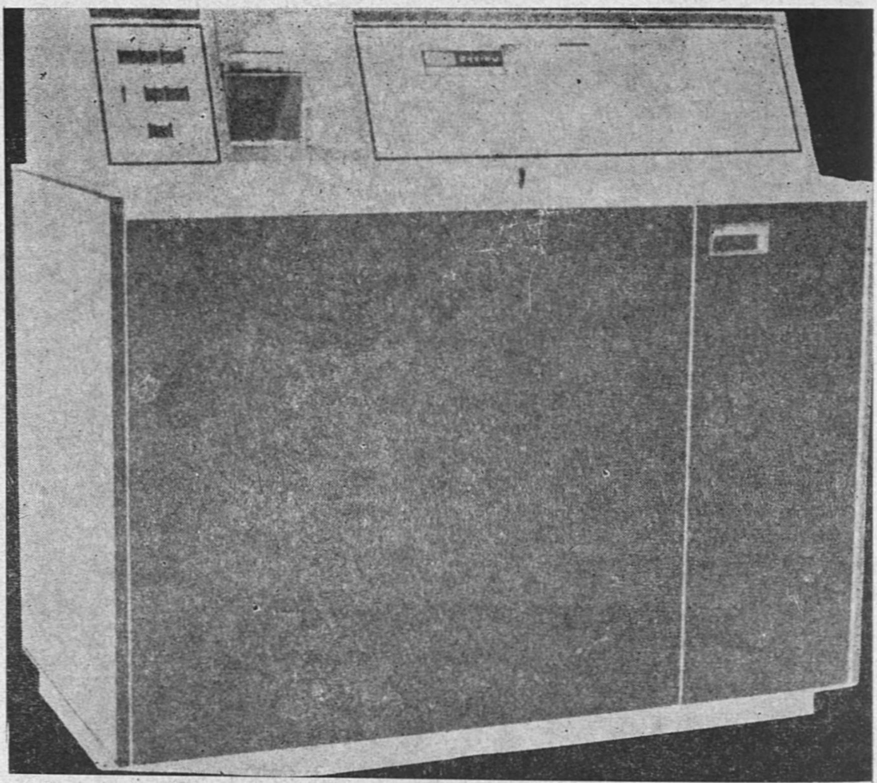
அட்டவணை 4 b



படம் 4.11
கால்குலேட்டர்



படம் 4.12
விளக்கி (Interpreter) IBM



படம் 4.12a

விளக்கி (யூனிவாக்)

மேற்கூறிய பலவிதமான அலுவல்களையும் கொலேட்டர் செய்யவல்லதாயினும் அவ்வலுவல்கள் ஒவ்வொன்றுக்கும் ஏற்ற வாறு கொலேட்டரின் மின்வழிச் சட்டம் (Panel) மாற்றியமைத்தல் நமது பொறுப்பாகும். இந்தச் சட்டத்தின் கட்டுப்பாட்டில் தான் கொலேட்டர் இயங்குகிறது; மின் வழி அமைப்புக்கு ஏற்றவாறு, ஒன்றல், தேர்ந்தொன்றல் ஆகிய பல வேலைகளையும் செய்கிறது.

கொலேட்டரின் வேகம் நிமிடத்திற்கு 240 முதல் 480 கார்டுகள் வரையாகும். இந்த வேகம் பெரும்பாலும் செய்யப் படும் வேலையைப் பொருத்ததாகும். கொலேட்டரின் ஒரு கார்டு அறை (பிரைமரி) மட்டுமே பயன்படும் வேலையாயின் அதன் வேகம் நிமிடத்திற்கு 240 கார்டுகளாம். கொலேட்டரின் இரண்டு படிப்பிடங்களும் பயன்படுத்தப்படும்போது, இவ்வேகம் நிமிடத் திற்கு 240-லிருந்து 480க்குள்ளான கார்டுகளாக மாறுகிறது.

கால்குலேட்டர் :— (Calculator)— கணிப்பொறி

யூனிட் ரெகார்ட் மெஷின்களில் சிறு கணிப்புக்களைச் (கூட் டல், கழித்தல், பெருக்கல், வகுத்தல்) செய்யும் பொறி ஒன்று உண்டு. அதுவே கால்குலேட்டர் எனப்படும் கணிப் பொறி ஆகும். ஒரு கார்டில் ஓரிடத்தில் ஒரு தொழிலாளியின் நாள் கூலியும் அதே கார்டில் பிறிதோர் இடத்தில் அவன் வேலை செய்த நாட்களின் எண்ணிக்கையும் துளைக்கப் பட்டிருப்பின் அந்தக் கார்டைக் கால்குலேட்டரில் இட்டு, அவனுக்குச் சேர வேண்டிய மொத்த கூலியைக் கணித்து, அதே கார்டில் மற்றொரு இடத்தில் அத்தொகையை துளைக்கலாம்; அல்லது மற்றொரு புதிய கார்டில் விரும்பிய இடத்தில் துளைக்கலாம். கால்குலேட்டரில் அதிகப்படி, நிமிடத்திற்கு 50 கார்டுகளே ஆய (Process) முடியும்.

விளக்கி :— (Interpreter)

ஒரு கார்டில் துளைக்கப்பட்டிருக்கும் செய்தியை அக்கார்டின் மீதே அச்சிடப் (Print) பயன் படும் பொறி விளக்கி (Interpreter) ஆகும். இவ்வாறு அச்சிடும் போது கார்டின் துளை களுக்கு நேர் மேலாக அதற்கான உரு (Character) அச்சிடப் படுவதில்லை; படிப்பதற்கு ஏற்றவாறு அவை இடைவெளியிட்டு, ஒரு வரிக்கு 60 உருக்கள் என்ற வீதத்தில் அச்சிடப்படு கின்றன. (ஒரு கார்டில் 80 துளையிடங்கள் உள்ளன என்பதை நினைவில் கொள்க.) அச்சிடும் முதல் வரி கார்டின் 12-ஆம்

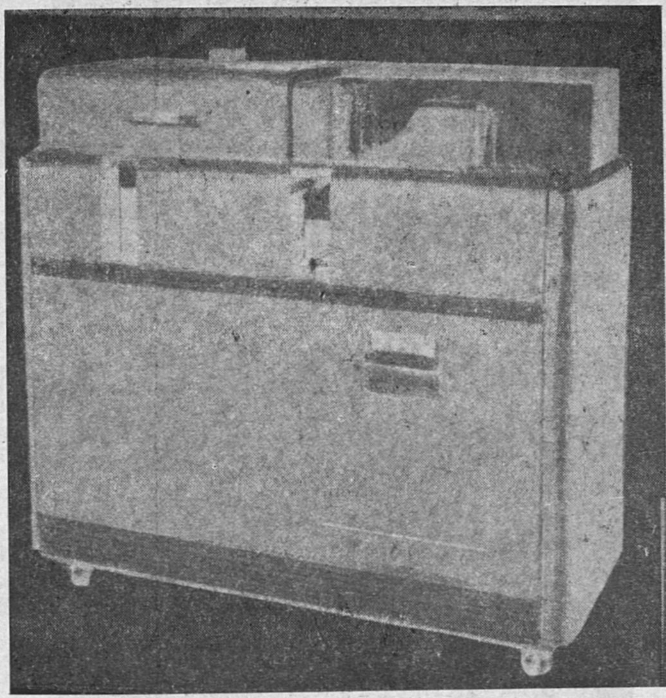
வரிசைக்குச் (12th Zone) சற்று மேலும், இரண்டாம் வரி 12, 11 வரிசைகளுக்கு இடையிலும் அமைந்து உள்ளன. தேவை யிருப்பின் கார்டில் துளையிடப்பட்டுள்ள விவரத்தை இரண்டு வரிகளிலும் இப்பொறி அச்சிடும். ஒவ்வொரு வரியை அச்சிடு வதற்கும் ஒரு முறை துளைக் கார்டை இண்டர்பிரட்டர் வழியாக செலுத்த வேண்டும். இண்டர்பிரட்டரின் வேகம் நிமிடத்திற்கு 60 கார்டுகளாம்.

விருத்திப்பொறி (Reproducer)

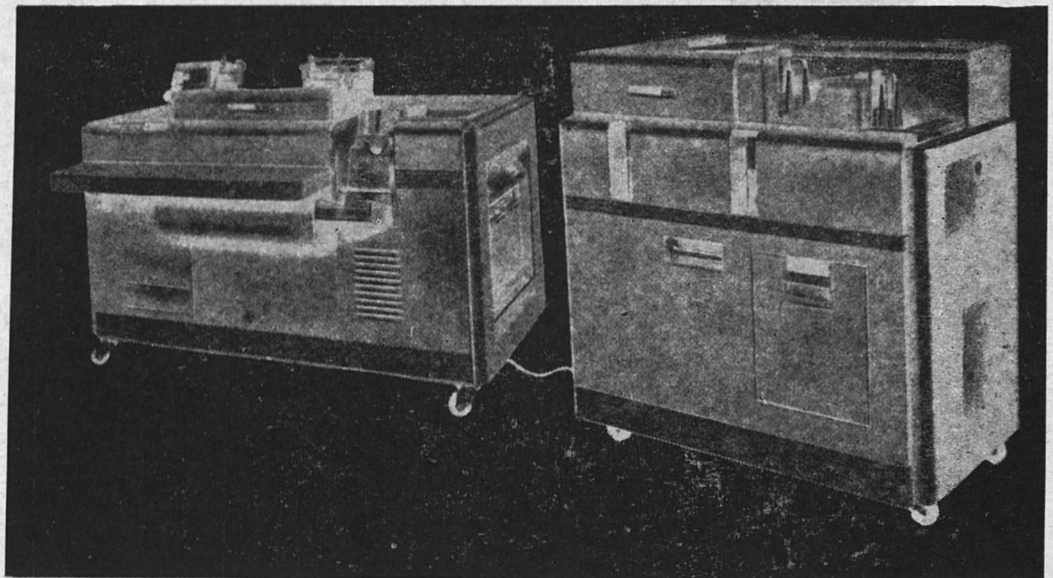
ஒரு கற்றைத் துளைக் கார்டுகளில் உள்ள விவரங்களை மற்றொரு கட்டுக் கார்டுகளில் கார்டுக்குக் கார்டு என்ற முறையில் பிரதியாகத் துளைக்க வேண்டிய அவசியம், அடிக்கடி ஏற்படும் ஒன்றும். இதைச் செய்ய விருத்திப்பொறி (Reproducer) என்ற ஒரு பொறியுள்ளது. இந்தப் பொறியைக் •கொண்டு, (i) ஒரு கார்டில் உள்ள முழு விவரங்களையும் அதன் முழுப் பிரதியாக மற்றொரு கார்டில் துளைக்க முடியும் (ii) ஒரு கார்டில் உள்ள விவரங்களின் ஒரு பகுதியை மட்டும் மற்றொரு கார்டில் துளைக்க முடியும். எடுத்துக் காட்டாக ஒரு கார்டில் முதல் நான்கு இடங்களில் ஒரு 4 இலக்க எண்ணும், 5-ஆம் இடத்திலிருந்து ஒரு நபரின் பெயரும் துளைக்கப்பட்டிருப்பின், ''விருத்திப் பொறியைக்'' கொண்டு மற்றொரு கார்டில், இந்த, 4 இலக்க எண்ணை மட்டும் (பிரதியாக) அதன் முதல் நான்கு இடங்களில் துளைத்துக் கொண்டு, முதல் கார்டில் இருந்த பெயரை விட்டு விட முடியும் (iii) ஒரு கார்டில் ஒரு பகுதியில் துளைக்கப்பட்ட விவரத்தை மற்றொரு கார்டில் பிரிதொரு பகுதியில் துளைக்க இயலும்.

எடுத்துக் காட்டாக, முதல் கார்டில் 1 முதல் 30 முடிய உள்ள இடங்களில் ஒரு பெயர் துளைக்கப் பட்டிருக்கிறதென்றால், விருத்தியைப் பயன் படுத்தி இந்த பெயரை மற்றொரு கார்டில் 17 முதல் 46 முடிய உள்ள இடங்களில் துளைக்கலாம்.

விருத்திப் பொறியில் வெற்று கார்டுகள் (துளையிடப்படாத கார்டுகள்) ஓரிடத்திலும், துளைக்கப்பட்ட கார்டுக் கட்டு மற்றொரு இடத்திலும் வைக்கப்பட்டு இருக்கும். துளைக்கப்பட்ட கார்டுகள் ஒன்றன் பின் ஒன்றாகப் படிப்பிடம் (Reading Station) ஒன்றின் வழியாகவும், வெற்றுக் கார்டுகள் துளைமேடை ஒன்றின் வழியாகவும் செல்லும். துளைக்கப்பட்ட கார்டு ஒன்று படிப் பிடத்தின் மேல் இருக்கும். அதே நேரத்தில், வெற்றுக்கார்டு ஒன்று துளை மேடை மீது இருக்குமாறு இப்பொறி இயங்கு கிறது. படிப்பிடத்தில் படிக்கப் பட்ட செய்தி துளை மேடைக்கு



படம் 4.14
விருத்திப் பொறி (Reproducer)



படம் 4.14a

அச்சுப் பொறியும் விருத்திப் பொறியும் இணைந்திருத்தல்

உடனடியாக அனுப்பப்பட்டு அங்குள்ள கார்டில் அது துளைக்கப் படுகிறது. துளைக்கப்பட்ட பின்னர் புதிதாகத் துளைக்கப்பட்ட கார்டு மீண்டும் படிப்பிடம் ஒன்றின் வழியாகச் செல்லும். இந்த கார்டில் உள்ள செய்தி இவ்விடத்தில் படிக்கப் பட்டதும் மூலக்கார்டில் படிக்கப் பட்ட செய்தியோடு ஒப்பு நோக்கப் படுகிறது. இரண்டும் முற்றிலும் ஒத்திருந்தால், வேலை தொடர்ந்து நடை பெறுகிறது. இரண்டிற்கும் வேறுபாடு இருப் பின் உடனே விருத்திப் பொறி நின்று, COMP (Compare) என்று குறிக்கப்பட்ட விளக்கு எரிகிறது.

முன் சொன்ன 3 காரியங்களைத் தவிர விருத்தி இன்னும் சில பணிகளையும் செய்கிறது. சில சமயங்களில் குறிப்பிட்ட ஒரு செய்தியை ஆயிரக்கணக்கான கார்டுகளில் துளையிடப்பட வேண்டியிருக்கலாம். இதற்கும் விருத்தி பயன்படுகிறது. இப் பணியில் குறிப்பிட்ட செய்தியைக் கொண்ட முதல் கார்டை நாம் துளைத்துக் கொள்ள வேண்டும். இந்தக் கார்டை முதல் கார்டாகக் கொண்டுள்ள ஒரு கற்றைக் கார்டுகள் துளைமேடை வழியாக ஒன்றன் பின் ஒன்றாக அனுப்பப்படுகின்றன. ஏற்கனவே துளைக் கப்பட்ட முதல் கார்டு (இது 'மாஸ்டர் கார்டு' என்று அழைக்கப் படும்) துளை மேடையைத் தாண்டி படிப்பு மேடையை அடையும் அதே நேரத்தில், மாஸ்டர் கார்டிற்கு அடுத்த கார்டு (வெற்று கார்டு) துளைமேடை மீது அமர்கிறது. படிப்பு மேடையில் உள்ள மாஸ்டர் கார்டு படிக்கப் பட்டு அதன் விவரம் அப்படியே துளை மேடைக்கு அனுப்பப்பட்டு அங்கு இப்போது காத்திருக்கும் வெற்றுக் கார்டில், மாஸ்டர் கார்டின் விவரங்களை, அதன் பிரதியாக துளைக்கப்படுகின்றன. இது முடிந்தவுடன் புதிதாக துளைக்கப்பட்ட கார்டு படிப்பு மேடைக்குச் செல்ல பழைய மாஸ்டர் கார்டு வெளியே வந்து விழுகிறது. படிப்பு மேடையில் இப்போதுள்ள (புதிதாக துளைக்கப்பட்டுள்ள) கார்டு, ஒரு மாஸ் டர் கார்டுபோல இயங்குகிறது. அதற்கு பின்புறம் வரும் வெற்றுக் கார்டு, துளை மேடைக்கு வரும்போது, புதிய மாஸ்டர் கார்டில் உள்ள விவரம் படிக்கப்பட்டு துளை மேடைக்கு அனுப் பப்பட்டு, மீண்டும் ஒரு கார்டு துளைக்கப்படுகிறது. இவ்வித தொடர் நிகழ்ச்சியால் (Chain action) அடுத்தடுத்து வரும் கார்டுகளில் முதல் கார்டில் காணப்பட்ட விவரங்களை தொடர்ந்து துளைக்கப்படுகின்றன. இதற்கு கூட்டுத்துளை (Gang Punch) என்று பெயர்.

சில சமயங்களில் விருத்திப் பொறி அக்கௌன்டிங் மெஷி னோடு (விவரம் பின்னால் காணப்படும்) இணைக்கப்பட்டிருக்கும்.

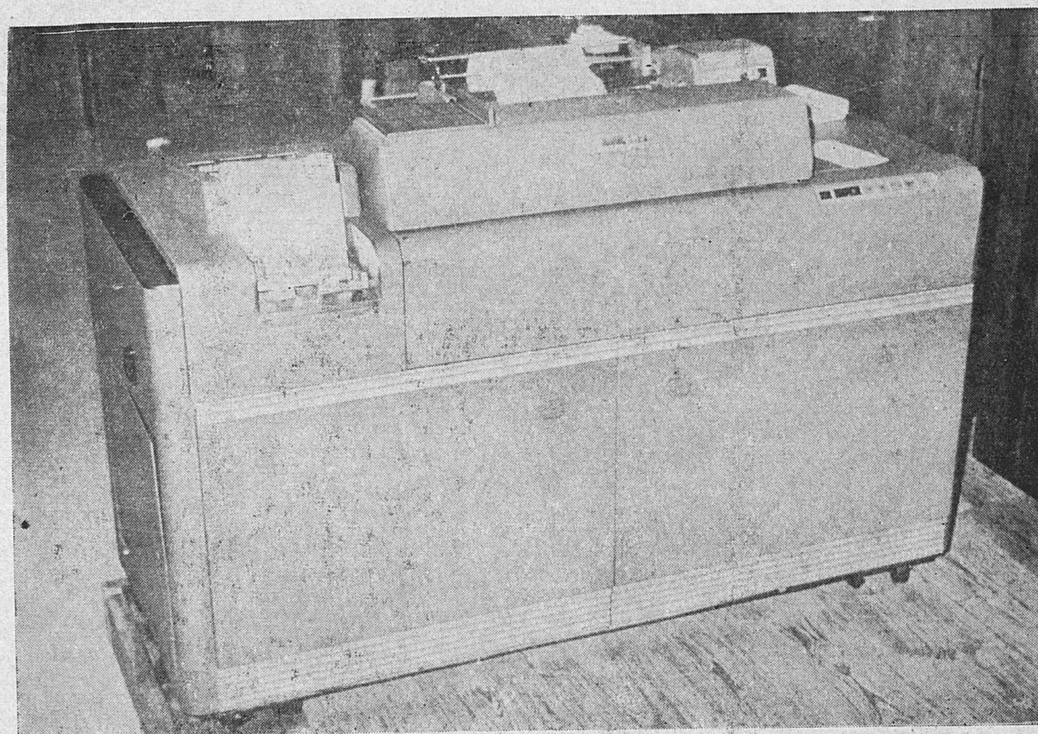
இவ்வித இணைப்பில் விருத்திப் பொறி அகௌண்டிங் மெஷினின் கட்டுப்பாட்டில் இயங்கும். அகௌண்டிங் மெஷின் இயங்கும் போது ஏற்படும் சுருக்க அறிக்கைகளை (Summary statements) அகௌண்டிங் மெஷினில் அச்சிடுவதுடன், கார்டுகளிலும் துளையிடவே இந்த இணைப்பு ஏற்படுத்தப்படுகிறது. சுருக்க அறிக்கையைத் தயாரித்தவுடன் அகௌண்டிங் மெஷின் அதை அச்சிடுவதுடன், அந்த விவரத்தை உடனே விருத்திப் பொறிக்கு அனுப்புகிறது. விருத்திப் பொறியும் அவ்விவரத்தைக் கார்டில் துளையிடுகிறது.

நிமிடத்திற்கு, சாதாரணமாக விருத்திப் பொறி 100 கார்டுகள் வீதம், படித்துத் துளைக்கிறது. அகௌண்டிங் மெஷினைப் போலவே விருத்திப் பொறியிலும் ஒரு மின் வழிப் பலகை (Wiring Panel) ஒன்று உள்ளது. இந்த பலகையில் நாம் விரும்பும் பணிக்கு ஏற்றவாறு இதிலுள்ள மின் இணைப்புகளை மாற்றி அமைக்க வேண்டும்.

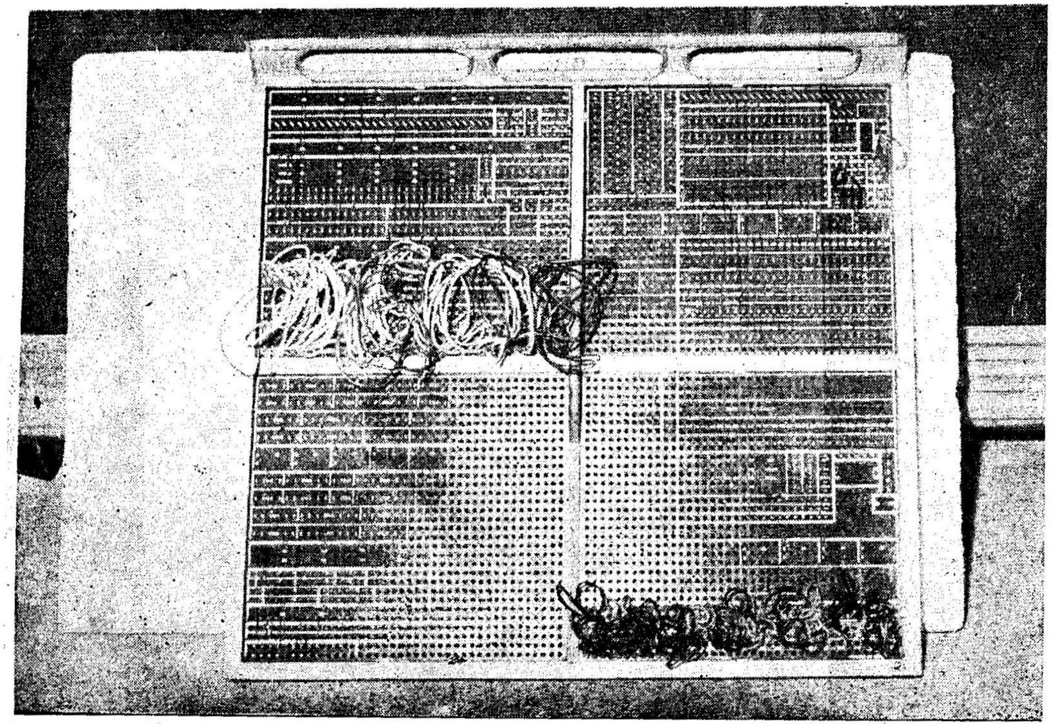
அக்கௌண்ட்டிங் மிஷின் (Accounting Machine)

துளைக் கார்டுகளைக் கொண்டு கணக்கு வைக்கும் முறை (Accounting) யில் முதல்படி துளைக் கார்டுகள் தயாரிப்பது; அடுத்தபடி தயாரிக்கப்பட்ட கார்டுகளை குறிப்பிட்ட எண் அடிப்படையில் ஏறும் அல்லது இறங்கும் வரிசையில் திருத்தி அமைப்பது; அதற்கும் அடுத்தபடி இவ்வாறு திருத்தி அமைக்கப்பட்டுள்ள கார்டுகளிலிருந்து அறிக்கை (Report) களைத் தயாரிப்பது. இத்தகைய அறிக்கைகள் பலதிறப்படும். ஒன்று முழுவிவர அறிக்கை (Detailed Report). இதன்படி ஒவ்வொரு கார்டில் காணும் செய்தியும் ஒவ்வொரு வரியில் அச்சிடப்படவேண்டும். அடுத்தது இடைநிலை அறிக்கை (Intermediate Report). இதன்படி ஒரு குறிப்பிட்டுள்ள குறியீடு உள்ள எல்லா கார்டுகளின் எண்களும் ஒன்ருகக் கூட்டப்பட்டு ஒரே வரியில் அச்சிடுவதாம்.

எடுத்துக்காட்டாக ஒரு பொருளின் விற்பனை விவரங்கள், மாநில, மாவட்ட விற்பனைமுகாம் வாரியாக அட்டவணை 4.1-ல் உள்ளதுபோல் கொடுக்கப்பட்டுள்ளது எனக் கொள்வோம். இவ் விவரங்களை ஒரு வரிக்கு ஒரு கார்டு என துளைத்தால் நமக்கு 10 கார்டுகள் கிடைக்கும். இந்தக் கார்டுக்கட்டை அப்படியே அக்கௌண்ட்டிங் மெஷினில் இட்டு அட்டவணை 4.1-ல் உள்ளது போல் ஒரு அட்டவணையை அச்சிட முடியும். இதையே நாம் முழுவிவர அறிக்கை (Detailed Report) என்று கூறுகிறோம். அடுத்தபடியாக இதே கார்டுகளை அக்கௌண்ட்டிங் மிஷினில் இட்டு ஒரு விற்பனை முகாமிற்கு ஒரு வரி மட்டுமே இருக்குமாறு

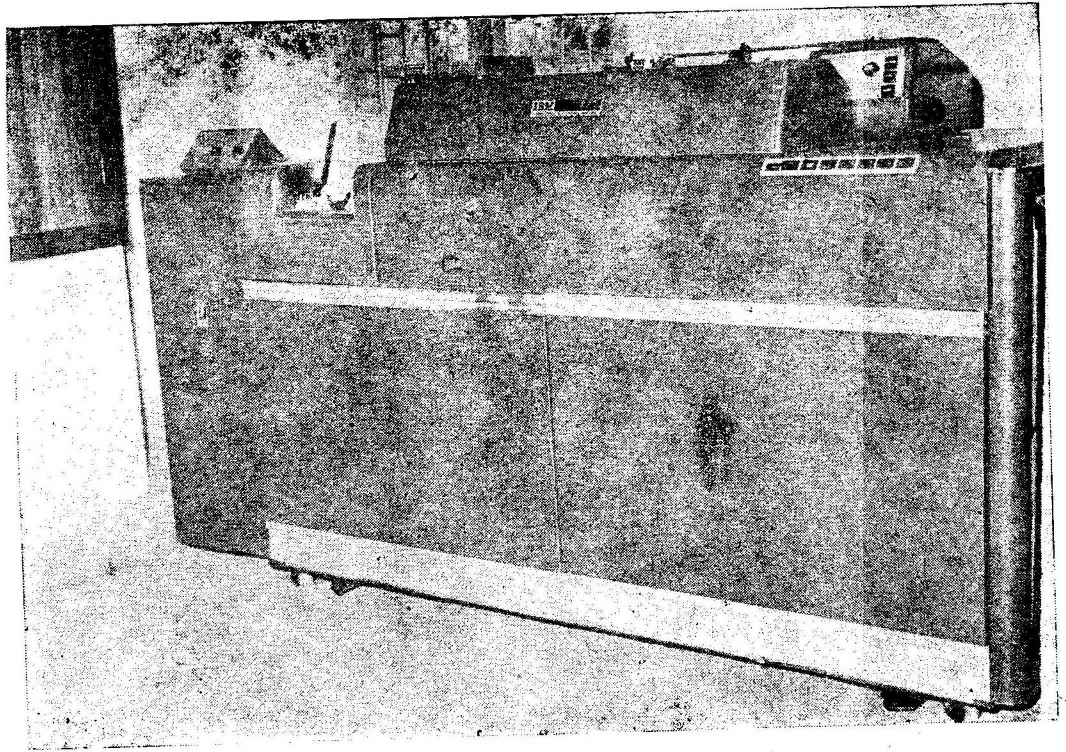


படம் 4.15



படம் 4.16

IBM 407 அகௌண்டிங் மெஷின் பயன்படுத்தும் மின்வழிச் சட்டம் **(Panel)**



படம் 4.17

IBM 447—அகௌண்டிங் மெஷின்

அச்சிடலாம். அதாவது ஒரே விற்பனை முகாமைச் சேர்ந்த கார்டு களில் உள்ள விற்பனைத் தொகைகள் கூட்டப்பெற்று ஒரே வரியில் கிடைக்குமாறும் அச்சிடலாம். (இதை இடைநிலை அறிக்கை எனலாம்).

அட்டவணை 4.2-ஐ காண்க. இதே முறையில் மீண்டும் அந்த கட்டுக் கார்டுகளை அக்கௌண்ட்டிங் மிஷினிலிட்டு அட்டவணை 4.2-ஐ மேலும் சுருக்கி மாவட்டவாரியாக (அட்டவணை 4.3) அச்சிடலாம். பின்னர் அதையும் சுருக்கி மாநிலவாரியாகவும் (அட்டவணை 4.4) அச்சிட இயலும். (இவ்வகை அறிக்கைகள் தனித்தனி விற்பனை நிலையங்களை, கண்காணிப்

முழுவிவர அறிக்கை

மாநில எண்	மாவட்ட எண்	விற்பனை நிலைய எண்	விற்பனை
6	2	3	1347.50
6	2	3	700.00
6	2	4	956.50
6	2	4	560.00
6	4	5	50.00
6	4	5	750.00
6	4	7	1350.00
8	5	9	2500.00
8	5	10	956.50
8	6	12	978.00
8	6	12	750.00

அட்டவணை 4.1

சுருக்க அறிக்கை (விற்பனை நிலையம் வாரியாக)

மாநில எண்	மாவட்ட எண்	விற்பனை நிலைய எண்	விற்பனை
6	2	3	2047.50
6	2	4	1516.50
6	4	5	800.00
6	4	7	1350.00
8	5	9	2510.00
8	5	10	956.00
8	6	12	1728.00

அட்டவணை 4.2

சுருக்க அறிக்கை (மாவட்டவாரியாக)

மாநில எண்	மாவட்ட எண்	விற்பனை
6	2	3564 00
6	4	2150.00
8	5	3466.00
8	6	1728.00

அட்டவணை 4.3

சுருக்க அறிக்கை (மாநில வாரியாக)

மாநில எண்	விற்பனை
6	5714.00
8	5194.00

அட்டவணை 4.4

போருக்கும் மாவட்ட விற்பனைத் தலைவருக்கும், மாநில விற்பனைத் தலைமைக்கும், கடைசியில் வியாபாரத்தின் உரிமையாளருக்கும் தேவைப்படும் என்பதை உணர்க) இத்தகைய அறிக்கைகளுக்கு 'சுருக்க அறிக்கைகள்' **(Summary Reports)** என்று பெயர்.

மேற்கூறிய பலவகையான அறிக்கைகளையும் தயாரிக்க அதற்கு உரிய முறையில் அக்கௌண்ட்டிங் மிஷினில் உள்ள மின்வழிச் சட்டத்தை **(Wiring Panel)** மாற்றியமைக்கவேண்டும்.

மேற்கூறிய பல வகையான அறிக்கைகளையும் தயாரிக்க அதற்கு உரிய முறையில் அக்கௌண்ட்டிங் மிஷினில் உள்ள மின்வழிச் சட்டத்தை **(Wiring Panel)** மாற்றியமைக்க வேண்டும்.

5

புரோகிராம் மொழிகள்

(Programming Languages)

ஒவ்வொரு கம்ப்யூட்டருக்கும் ஒரு பொறி மொழி (Machine Language) ஒன்று உண்டு என்றும், இம்மொழி கம்ப்யூட்டரின் வடிவமைப்பிலேயே பிறந்ததென்றும் முன்னர் பார்த்தோம். பெரும்பாலும் இத்தகைய மொழிகள் வெரும் எண்களின் கோர் வையாகவே காணப்படும். இத்தகைய மொழிகள் புரொகிராம் எழுதுவது, அதுவும் விஞ்ஞானம், பொறியியல் போன்ற துறை களில் தேவைப்படும் கணிப்புகளுக்கான புரோகிராம்களை எழுது வது என்பது மிகவும் கடினவான ஒன்றும். அப்படியே சிரமப் பட்டு எழுதி முடித்தபோதிலும் அதன் பின்னர் அதில் ஏதேனும் பிழை யிருப்பின் கண்டுபிடிப்பது என்பதோ அல்லது நாம் வெற்றிகரமாகப் பயன்படுத்திக்கொண்டிருக்கும் புரோகிராமில் அப்போதைக்கப்போது மாற்றங்கள் செய்வதோ அவ்வளவு எளி தான காரியம் அல்ல. பொறி மொழியில் எழுதப்படும் புரோ கிராமில் ஆணைகள் யாவும் கம்ப்யூட்டரின் செயல் திறன்களுக்கு உட்பட்டே இருக்கவேண்டுமாதலின், அத்தகைய புரோகிராம் மிக விரிவானதாகவும், பெரியதாகவும் இருக்கும். இதனல் இத் தகைய புரோகிராம் ஒன்றை எழுத பெரும் காலச் செலவும் ஆகும். சில வகையான கணக்குகளுக்குப் பொறி மொழியில் புரோகிராம் எழுத நினைத்தால் அதற்கு பல மாதங்கள் தேவைப் படலாம்.

இத்தகைய தடைகளால், கம்ப்யூட்டர்கள் தோன்றியவுடன் அவற்றிற்கு இன்றுள்ள பரந்த வரவேற்போ அல்லது அவற்றை பெருத்த அளவில் பயன்படுத்தும் நிலையோ ஏற்படவில்லை. கம்ப்பைலர் (Compiler) என்ற மொழி பெயர்ப்பு வசதிகள் ஏற்பட்ட பின்னரே கம்ப்யூட்டர்கள் பெருமளவிற்கு வரவேற்கப்பட்டன. இந்த கம்ப்பைலர் என்பன என்னவென்று பார்ப்போம். இதற்கு ஒரு சிறிய எடுத்துக்காட்டை பயன்படுத்துவோம்.

முதல் அத்தியாயத்தில் ஒரு சிறுவனைக் கொண்டு புத்தகம் வாங்கி வரும் உதாரணம் ஒன்றைப் பார்த்தோம். நமது சிறுவனுக்குரிய கட்டளைகளை நாம் இடும்பொழுது, நாம் பயன் படுத்தும் மொழி சிறுவனுக்குத் தெரிந்த மொழியாக இருக்க வேண்டும். அவனுக்கு அவனது தாய்மொழி மட்டும்தான் தெரியும் என்று கொண்டால் அந்த மொழியை நாம் கற்றுக் கொண்ட பின்னரே, அம்மொழியில் அவனுக்கு நாம் ஆணைகளை இடமுடியும். சிறுவனின் மொழி, கற்பதற்கு மிகவும் கடினமான தாக இருந்து அதைக் கற்றுக் கொள்வதும் தொல்லை நிறைந்த தாய் இருந்தால் என்ன செய்வது? ஒன்று செய்யலாம். வேறு ஒரு புதிய ஆளின் உதவியை நாடலாம். நமது சிறுவனின் மொழியைவிட இந்தப் புதிய ஆளின் மொழி கற்றுக்கொள்ள எளிதாகவும், கையாளுவதற்குச் சௌகரியமாகவும் இருப்பின், இப் புதிய ஆளின் மொழியைக் கற்றுக்கொள்வோம்; கற்றுக் கொண்ட பின்னர் சிறுவனுக்கு இடவேண்டிய கட்டளைகளைப் புதியவரின் மொழியில் தயாரித்து, அவரிடம் கொடுத்து, அவற்றை மொழிபெயர்த்து சிறுவனுக்குக் கூறும்படி, ஏற்பாடு செய்துகொள்ளலாம். இந்த ஏற்பாட்டில் சிறிது கால விரயம் (மொழிபெயர்ப்புக்கான அதிகப்படி காலம்) ஆகக்கூடும். ஆனால் கிடைக்கக்கூடிய சௌகரியத்தை ஒப்பு நோக்கும்போது இந்தக் கால தாமதம் பொறுத்துக்கொள்ளக் கூடியதே.

மேற்சொன்ன எடுத்துக் காட்டில் கூறப்பட்டதைப்போல கம்ப்யூட்டரின பொறி மொழியில் ஆணைகளை எழுதிப் புரோகிராம் தயாரித்துப் பயன்படுத்துவதில் ஏற்படும் தொல்லைகளைத் தவிர்க்க, மொழிபெயர்ப்பாளர்களை உண்டாக்கியுள்ளார்கள். இந்த மொழி பெயர்ப்பாளர்களை 'கம்பைலர்கள்' என்று கூறுவார்கள். (உண் மையில் இவை வெறும் மொழிபெயர்ப்புகளை மட்டும் செய்வ தில்லை. மேலும் பல வேலைகளையும் செய்கின்றன.)

கம்பைலர் என்பது பெரிய புரோகிராமே யாகும். இந்தக் கம்பைலர் பொறி மொழியில்தான் எழுதப்பட்டிருக்கும். நாம் நமது புரோகிராமை 'உயர் நிலைமொழி' (Higher Level Language) என்ற மொழியில் எழுதிக் கம்ப்யூட்டருக்குக் கொடுக் கிறோம். இந்த மொழி மற்ற மொழிகளைப்போலவே இலக்கணக் கட்டுப்பாட்டுக்கு அடங்கியே இருக்கும். இந்த உயர்நிலை மொழி யில் உள்ள கட்டளைகளைக் கம்பைலர் பொறி மொழியில் மொழி பெயர்த்துக் கம்ப்யூட்டருக்கு அளிக்கும்.

சில சமயம் உயர்நிலை மொழிக் கட்டளைகள் கம்பைலரால் அசெம்பிளி மொழி (Assembly Language) என்ற புதிய மொழி

யில் மொழிபெயர்க்கப்படும். பிறகு இந்த அசெம்பிளி மொழிக் கட்டளைகள் 'அசெம்பளர்' என்ற மற்றொரு புரோகிராமால் பொறி மொழியில் மொழிபெயர்க்கப்படும். அசெம்பிளி மொழி பொறி மொழியைக்காட்டிலும் எளிதானதாக இருக்கும். இருப்பினும் அசெம்பிளி மொழி உயர்நிலை மொழியினைப் போல அவ்வளவு எளிதானவை அல்ல. கீழுள்ள மூன்று எடுத்துக் காட்டுகளைக் காணவும்.

```
3607500    00100
3807500    00100
48
```

படம் 5.1

பொறி மொழி ஆணைகள் : எடுத்துக்காட்டு

```
DS   PAY
A    PAY, DA
S    PAY, TAX
  ...
  ...
```

படம் 5.2

ஓர் அசெம்பிளி மொழி ஆணைகள் : எடுத்துக்காட்டு

```
      A : 15.5
      B : 17
      C : A+B
PRINT,   A, B, C.
      END
```

படம் 5.3

ஓர் உயர்நிலை மொழி ஆணைகள் : எடுத்துக்காட்டு

இந்த மூன்று அட்டவணைகளை ஒப்பு நோக்குங்கள். முதல் அட்டவணையிலிருந்து ஒன்றுமே புரிந்துகொள்ள முடியவில்லை அல்லவா? இரண்டாவது அட்டவணைக்கு வாருங்கள். அந்த அட்டவணையிலுள்ள ஆணைகள் எந்தப் பணியைச்செய்ய எழுதப் பட்டிருக்கின்றன என்பது தெளிவாக ஒன்றும் தெரியவில்லை; ஆனால் ஏதோ PAY, DA, TAX என்ற வார்த்தைகள் இருப்ப தால் அது ஒருவரின் சம்பளத்தைக் கணக்கிடப் பயன்படுத்தப் படும் புரோகிராமின் ஒரு பகுதியாக இருக்கலாம்போலத் தோன்று கிறது. அட்டவணை 3-ஐப் பார்த்தால் முற்றும் தெளிவாகத் தெரிகிறது. (எந்த விதமான சிக்கலுமின்றிப் பள்ளிச் சிறுவனும் புரிந்துகொள்ளக்கூடிய அளவில் தெளிவாக இருக்கிறதல்லவா? சரி இந்த புரோகிராம் என்ன செய்கிறது?) A, B இவற்றின் மதிப்புகள் கொடுக்கப்பட்டு உள்ளன. அவற்றைப் பயன்படுத்தி C-ன் மதிப்பை C=A+B என்ற சமன்பாட்டைப் பயன்படுத்திக் கண்டுபிடிக்க வேண்டும். பின்னர் A, B, C ஆகியவற்றின் மதிப்புகளை அச்சிட வேண்டும்.

ஓர் அசெம்பிளி மொழியில் எழுதப்பட்ட புரோகிராம் ஒன்றில் காணும் ஆணைகளின் எண்ணிக்கை பெரும்பாலும் அதே புரோ கிராமின் பொறி மொழி ஆணைகளின் எண்ணிக்கைக்குச் சமமாக இருக்கும். இரண்டு புரோகிராம்களுக்குமிடையே ஒன்றுக்கு ஒன்று என்ற விகிதத்திலேயே (one to one correspondence) ஆணைகள் உள்ளன. பொறி மொழியை ஒப்புநோக்க. அசெம்பிளி மொழி யில் உள்ள வசதி, புரோகிராம்களில் குறிப் பெயர்களைப் (Mnemonics) பயன்படுத்தக்கூடிய சௌகரியமே. அட்டவணை 2-ல் PAY என்றும் DA என்றும் TAX என்றும் காணும் வார்த் தைகள் உண்மையிலேயே ஒருவரின் அடிப்படைச் சம்பளம், கிராக்கிப்படி, வரி ஆகியவற்றிற்கான தொகைகளை முறையே குறிக்கலாம். A என்பது ADD கூட்டு என்பதற்கும், S என்பது 'கழி' (Subtract) என்பதற்குமான குறிப் பெயர்களாம். ஆனால் உயர்நிலை மொழியில் இந்த சௌகரியங்களைவிட மேலும் பல அடிப்படையான சௌகரியங்கள் உண்டு. அவற்றுள் முக்கிய மானது, பொறி மொழியில் எழுதவேண்டிய ஒரு கொத்து ஆணை களுக்கு இணையாக (சில சமயங்களில் 100-க்கு மேற்பட்ட பொறி ஆணைகள் கூட இருக்கலாம்) ஒரே ஓர் ஆணையை உயர்நிலை மொழியில் அளிக்கலாம். இதற்கு ஓர் எடுத்துக்காட்டு வேண்டு மாயின் நமது மொழிபெயர்ப்பாளரையும், சிறுவனையும் மீண்டும் அழைக்கவேண்டும். புத்தகம் வாங்கிவருவதற்கு வேண்டிய 6 ஆணைகளைச் சிறுவனுக்கு நாம் நேரிடையாக அளித்தோமல்லவா? அதற்கு மாறாக இப்போது மொழிபெயர்ப்பாளரிடம் நம் மொழி

யில் 'புத்தகம் வாங்கிவா' என்றுமட்டும் கூறுவோம். உடனே மொழிபெயர்ப்பாளர் வேலையாளின் மொழியில்,

(i) வீட்டு வாசலுக்குப்போய்த் தெருவில் இறங்கு.
(ii) உன் இடப்புறமாக 200 மீட்டர் நடந்து செல்.
(iii) தெருவைக் குறுக்காகக் கடந்துகொள்.
(iv) கடந்தவுடன் உன் இடப்புறம், மூன்றும் நுழை வாயிலுக்குச் செல்.
(v) புத்தகத்தை வாங்கிக்கொள்.
(vi) சென்ற வழியே திரும்பி வா.

ஆகிய ஆறு ஆணைகளையும் தானே உண்டாக்கி அவற்றைச் சிறுவனுக்கு அளிக்கிறார்.

பொறி மொழியின் ஆணைகளைச் சிற்றாணைகள் (Micro Instructions) என்றும், உயர்நிலை மொழியில் எழுதப்பட்டுள்ள ஆணைகளைப் பேராணைகள் (Macro Instructions) என்றும் அழைப்பர்.

ஒவ்வொரு கம்ப்யூட்டருக்கும் ஒரு பொறி மொழி இருப்பது போலவே அவற்றிற்கு ஓர் அசெம்பிளி மொழியையும் தயாரித்திருப்பார்கள். இவற்றைப் பொதுவாகக் குறியீட்டுப் புரோகிராம் மொழி (Symbolic Program Language) என்றும் அழைப்பர்.

உயர்நிலை மொழிகள் பல உள்ளன. FORTRAN, ALGOL, COBOL, PL/1 இவை உயர்நிலை மொழிகளில் சிலவாகும்.

உயர்நிலை மொழிகள் சாதாரணமாக எந்த ஒரு தனிப்பட்ட கம்ப்யூட்டருக்காகவும் எழுதப்படுவதில்லை. ஓர் உயர்நிலை மொழியைப் பல வேறுவகையான கம்ப்யூட்டர்களிலும் பயன்படுத்தலாம். எடுத்துக்காட்டாக FORTRAN-ஐ எடுத்துக்கொண்டால் இன்றைய தலைமுறை கம்ப்யூட்டர்கள் யாவும் இந்த மொழியைப் பயன்படுத்தும் விதத்தில் உள்ளன. எந்த ஒரு கம்ப்யூட்டராயினும் FORTRAN அல்லது அதற்கு இணையான ஓர் உயர்நிலை மொழியைப் பயன்படுத்த இயலாததாயின் அதன் தரம் குறைந்ததாகவே இன்று கருதப்படுகிறது.

மனிதர்கள் பேசும் ஆயிரக்கணக்கான மொழிகளில், ஒரு சில மொழிகளே முக்கியத்துவம் வாய்ந்தவையாகவும், பரவலாகப் பயன்படுத்தப்படும் நிலையிலும் உள்ளனவோ, அவ்வாறே கம்ப்யூட்டர் உலகிலும் நூற்றுக்கணக்கான உயர்நிலை மொழிகள் இருப்பினும் ஒரு சில உயர்நிலை மொழிகளே பெருமளவில் பயன்படுத்தப்படுகின்றன.

ஃபோர்ட்ரான் (FORTRAN)

இந்த அத்தியாயத்தின் எஞ்சியுள்ள பகுதியில் ஒரு புரோ கிராம் எப்படி வேலை செய்கிறது என்பதைப்பற்றித் தெரிந்து கொள்ளுவோம். இதற்கு முன்னோடியாக ஃபோர்ட்ரான் (FORT-RAN) என்ற உயர்நிலை மொழி (High Level Language) ஒன்றின் அடிப்படைகளைக் கற்போம். கீழ்வரும் பத்திகளில் ஃபோர்ட்ரான் மொழி முழுவதையும் விவரிப்பது இப் புத்தகத்தின் நோக்கம் அன்று. புரோகிராம் என்றால் என்ன என்பதைத் தெரிந்து கொள்ளப் போதிய அளவே இங்கு விவரிக்கப்படும்.

ஃபோர்ட்ரானின் உருக்கள் (FORTRAN Characters)

எந்த ஒரு மொழிக்கும்—அது மனிதர்களின் மொழியாகட்டும் அல்லது கம்ப்யூட்டரின் மொழியாகட்டும்—அதன் அடிப்படை, எழுத்துகளும், எண்களும், சில சிறப்புக் குறியீடுகளும் ஆகும். ஃபோர்ட்ரானின் உருக்கள் ஆங்கில மொழியின் A முதல் Z முடிய உள்ள 26 எழுத்துகளும் 0, 1, 2...9 இலக்கங்களும் $, ', +, —, *, /, (, ¹,), =, ·, ஆகிய சிறப்புக் குறிகளும் ஆகும். இந்த உருக்களைக் கார்டுகளில் துளைக்கும்போது ஏற்படும் கார்டின் துளைக் குறிகள் அட்டவணை 5.1-ல் கொடுக்கப் பட்டுள்ளன.

ஃபோர்ட்ரான் எண்கள்

ஃபோர்ட்ரான், எண்களை முழு எண்கள் என்றும், தசம பின்ன எண்கள் என்றும் பிரிக்கிறது. எந்த எண்ணிலாவது தசமப் புள்ளி காணப்பட்டால் அது தசம எண் ஆகும்; தசம புள்ளி அற்ற எண் முழு எண் ஆகும்.

எடுத்துக்காட்டு :

முழு எண்கள்:

00
279
— 478
1234

தசம பின்ன எண்கள்:

1 ..24869
— 151 . 3250
1234 .
00 .

உரு	துளைச் சேர்க்கை	உரு	துளைச் சேர்க்கை
A	12,1	0	0
B	12,2	1	1
C	12,3	2	2
D	12,4	3	3
E	12,5	4	4
F	12,6	5	5
G	12,7	6	6
H	12,8	7	7
I	12,9	8	8
J	11,1	9	9
K	11,2	(	0,4,8
L	11,3	)	12,4,8
M	11,4	*	11,4,8
N	11,5	(வியங்கோள்) '	4,8
O	11,6	(கமா) ,	0,3,8
P	11,7	$	11,3,8
Q	11,8	*	12,3,8
R	11,9	=	3,8
S	0,2	—	11
T	0,3	+	12
U	0,4	/	0,1
V	0,5		
W	0,6		
X	0,7		
Y	0,8		
Z	0,9		

அட்டவணை 5.1

சில ஃபோர்ட்ரான் உருக்களும் அவற்றின் கார்டு துளைச் சேர்க்கைகளும்

(ஃபோர்ட்ரான் மட்டுமின்றி இன்னும் பல உயர்நிலை மொழிகளும் இக்குறியீடுகளைப் பயன்படுத்துகின்றன.)

மேலுள்ள எடுத்துக்காட்டிலிருந்து நாம் முக்கியமான ஒன்றை நினைவில் வைத்துக்கொள்ளுவோம். ஃபோர்ட்ரானில் ஒரு முழு எண் தசம புள்ளி இல்லாமலிருந்தால் அது ஒரு முழு எண்ணாக வும், அதற்கே ஒரு தசம புள்ளி வைத்துவிட்டால் அது ஒரு தசம எண்ணாகவும் ஆகிவிடுகிறது.

ஃபோர்ட்ரானின் சொற்கள்

ஒரு மொழியின் எழுத்துகளுக்கு அடுத்த நிலை, 'சொற்கள்' ஆகும். எந்த மொழியிலும், எழுத்துகளைக் கோவையாக்கி, சொற்களை உண்டாக்குவதில் சில வரைமுறைகளும், விதிகளும் உள்ளன. ஃபோர்ட்ரானிலும் அவ்வாறே. ஃபோர்ட்ரானின் சொற் களை இரண்டு வகைகளாகப் பிரிக்கலாம். ஒன்று கம்ப்யூட்டர் செய்யவேண்டிய பணிகளைக் குறிக்கும் வினைச்சொற்கள் (அட்ட வணை 5.2-ஐ காண்க); மற்றது, பெயர்ச் சொற்கள். பெயர்ச் சொற்கள் என்பது சாதாரணமாகக் கணிப்பில் பயன்படுத்தப் படும் குறியீடுகளாம். முன் பார்த்த படம் 5.3-ல் **A, B, C**

வரிசை எண்	ஃபோர்ட்ரானின் சில வினைச் சொற்கள்
1	ACCEPT
2	CALL
3	CONTINUE
4	ASSIGN
5	DO
6	END
7	WRITE
8	GOTO
9	PAUSE
10	PRINT
11	PUNCH
12	READ
13	STOP
14	TYPE

அட்டவணை 5.2

ஆகியவை பெயர்ச் சொற்களும் PRINT என்பது ஒரு வினைச் சொல்லும் ஆகும்.

பெயர்ச்சொல் விதிகள்

ஃபோர்ட்ரானின் பெயர்ச் சொற்கள் புரோகிராம் தயாரிப் போரால் (புரோகிராமர்) அவரது வசதிக்கு ஏற்ப தயாரித்துக் கொள்ளும் சொற்களாம். உதாரணமாக: ஓர் ஊதியக் கணிப்பில் ரூ. 1850 சம்பளமாக (PAY) இருக்கிறதென்றால் அதை PAY என்ற பெயராலேயே குறிப்பிட இயலும். கம்ப்யூட்டர் இந்தப் பெயரையும் அதன் 'மதிப்பான' 1850 என்ற எண்ணையும் வேண்டும்போது இணைத்துக்கொள்ளும். இந்த முறையில்தான் படம் 5.3-ல் C = A + B என்பதைக் கணிக்க, 'A' சொல்லின் மதிப்பையும் (இனிமேல் பெயர்ச்சொல் என்று கூறாமல் 'சொல்' என்றே கூறுவோம்) 'B' என்ற சொல்லின் மதிப்பையும் (A = 15.5, B = 17) முன் வரிசைகளிலிருந்து கம்ப்யூட்டர் பெற்றுக்கொள்ளுகிறது.

இனி சொற்களை உருவாக்குவதிலான விதிகளைக் காண்போம்:

விதி (1) ஒரு சொல்லில் 6 குறியீடுகளுக்குமேல் இருத்தலாகாது.

விதி (2) முதல் குறியீடு எப்போதும், A முதல் Z வரையிலான எழுத்துகளில் ஒன்றாகத்தான் இருத்தல் வேண்டும்.

விதி (3) சிறப்புக் குறியீடு ஏதும் பயன்படுத்தலாகாது.

எடுத்துக்காட்டுகள் :

சரியான சொற்கள்

(i) X
(ii) X X X X
(iii) PAY
(iv) JOE
(v) TAX
(vi) X 1 2 3 4
(vii) ENTERT
(viii) CHARGE

(மேலுள்ளவற்றுள் (vi) ஆவது எடுத்துக்காட்டில் எண்கள் பயன்படுத்தப்பட்டிருப்பதையும், இது மேற்சொன்ன (1) — (3) விதிகளை மதிக்கின்றது என்பதையும் காண்க.)

தவருன சொற்கள்

X / Y = (/ என்ற சிறப்புக் குறியால் 3ஆம் விதி மீறப்படுகிறது.)

72 XYZ = (2ஆம் விதி மீறப்பட்டுள்ளது.)
MEDIATE = (முதல் விதி மீறப்பட்டுள்ளது.)

ஒவ்வொரு பெயர்ச் சொல்லுக்கும் ஓர் எண் மதிப்பு (value) உண்டு என்று பார்த்தோம். படம் 5.3-ல் A-க்கு எண் மதிப்பு 15·5-ம், B-ன் மதிப்பு 17-ம், C-ன் மதிப்பு 32·5-ம் (C-ன் மதிப்பை கணித்த பின்னர்) ஆகும்.

ஃபோர்ட்ரானின் விதிகளின்படி எண்கள் இரண்டு வகை யானவை (முழு எண், தசம பின்ன எண் ஆகிய இரண்டு) என்றும், பெயர்ச்சொல் ஒவ்வொன்றுக்கும் ஓர் எண் மதிப்பு உண்டென்றும் பார்த்தோம். அப்படியானல் பெயர்ச் சொற்களும் இரண்டு வகையாகப் பிரிக்கப்படவேண்டும் என்று ஆகிறது. முழு எண்களைக் குறிக்கும் பெயர்ச் சொற்கள், தசம பின்ன எண்களைக் குறிக்கும் சொற்கள் என அவை பிரிக்கப்படல் வேண்டும். இதற்கு ஏற்றவாறு வேறு ஒரு விதியும் உண்டு.

முழு எண்ணைக் குறிக்கும் பெயரில் முதல் உரு I, J, K, L, M, N ஆகியவற்றுள், ஏதாவது ஒன்ருக இருத்தல் வேண்டும். முதல் எழுத்து இந்த ஆறு எழுத்துகளைத் தவிர, பிற எழுத்து களாயின் அச்சொல் தசம பின்ன எண்ணைக் குறிக்கப் பயன் படும். இவ்விரு சொற்களையும் முறையே 'முற்றெண் சொல்' 'பின்னச் சொல்' என்று அழைப்போம்.

உதாரணங்கள் :

முற்றெண் சொற்கள் = JACK
KIS
KXYZ
LONG
MMMMM
NOTE
IOTAAS

பின்னச் சொற்கள் ≡ XETA
YZ 250
TIMBER
WORK
AUGUST
B
C 5

கணிப்புக் குறிகள்

கம்ப்யூட்டர்களின் அடிப்படை வேலை கணிப்பதாகும். ஆகவே ஃபோர்ட்ரானின் கணிப்பிற்கான குறிகள் இருத்தல் அவசியமாகிறது. கீழ் வரும் கணிப்புக் குறிகளை ஃபோர்ட்ரான் பயன்படுத்துகிறது.

தொடர் (Expression)

எண்களையும், சொற்களையும், கணிப்புக் குறியீடுகளையும் கொண்டு ஏற்படுத்துவது தொடர் (Expression) ஆகும்.

எடுத்துக்காட்டுகள்:

(i) 15.75 * 17.321 /B + TT * * Y2
(ii) JACK + JIM — LOAD * IKEN * * M
(iii) T + 15.73
(iv) JJ + KK
(v) X * * Y
(vi) J * * K

மேற்காட்டிய தொடர்களைத் தயாரிக்கும்போது கவனத்தில் கொள்ளவேண்டியதொன்று உண்டு. ஒரு தொடர் முற்றும் முற்றெண் நிலையிலேயோ (Integer Mode) அல்லது முற்றும் பின்ன நிலையிலேயோ (Real Mode) இருத்தல் வேண்டும். வெவ்வேறு நிலைகளில் (Modes) உள்ள எண்களையும், சொற்களையும் இணைத்து எழுதப்படும் தொடரை ஃபோர்ட்ரான் அனுமதிப்பதில்லை.

மேலுள்ள எடுத்துக்காட்டுகளில் (i) (iii) (v) ஆகியவை பின்ன நிலைத் தொடர்களாம். மற்றவை முற்றெண் தொடர்களாம். அனுமதிக்கப்படாத தொடருக்கு எடுத்துக்காட்டு.

JACK + 15.5

குறி	செயல்	எடுத்துக்காட்டு
+	கூட்டல்	(i) 47.35 + 75.2 (ii) AMOUNT + 15.73 (iii) PAY + DA (iv) J + K2
—	கழித்தல்	(i) 76.125 — 1.32645 (ii) PAY — TAX (iii) TOTAM — 193.75 (iv) KOTA — LOTA
*	பெருக்கல்	(i) 75 * 153 (ii) 176.2 * FAT (iii) BIG * CAM
/	வகுத்தல்	(i) A / B (ii) 17.3364 / 44.3259 (iii) 96.35 / CA (iv) TIP / PEDA (v) TOKEN / 15.75
* *	படி உயர்த்தல் Exponentiation	(i) 14 * * 15 (= 14^{15}) (ii) TOM * * 16.3 (iii) 17.65 * * YOTT (iv) BAS * * STOK

JACK என்பது முற்றெண் சொல், 15.5 ஒரு **பின்னச்சொல்** என்றிருப்பதால் மேற்சொன்ன தொடரில் கலப்பு ஏற்பட்டு அது அனுமதிக்கப்படாததாகிறது.

ஃபோர்ட்ரான் தொடரில் கணிப்பு வரிசை முறை: ஒரு தொடரை எடுத்துக்கொண்டால், அத் தொடரில் உள்ள கணிப்புகள் எந்த வரிசையில் நடைபெறுகின்றன என்பதை நாம் அறிந்துகொள்ளல் அவசியம். அப்போதுதான் நாம் கணிக்க வேண்டிய பெரிய கணித சூத்திரங்களையும் எவ்வாறு ஃபோர்ட்ரான் தொடராக (Fortran Expression) எழுதவேண்டும் என்பதும் நமக்குத் தெரியும்.

மனிதர்களைப் போலவே கம்ப்யூட்டரும் ஒரு சமயத்தில் ஒரே ஒரு கணிப்பை மட்டுமே செய்யவல்லது. எடுத்துக்காட்டாக 17.5 + 6.7 — 6.2 என்ற தொடரில், நாம் முதன் முதலாகக் கூட்டலைச் செய்து அதன் **பின்னர்க்** கழித்தலைச் செய்வோம்; அல்லது கழித்தலைச் செய்த **பின்னர்க்** கூட்டலைச் செய்கிறோம். இவ்வாறே கம்ப்யூட்டரும் எவ்வளவு பெரிய தொடராயினும், அதில் உள்ள கணிப்புகளை ஒரு நேரத்தில் ஒரு கணிப்பு என்ற முறையில்தான் செய்கிறது. அவ்வாறு செய்யும்போது ஒரு தொடரில் எந்தக் கணிப்பை முதலில் செய்யவேண்டும், எதைப் பின்னர் செய்யவேண்டும் என்பதற்கான ஒரு வரைமுறை தேவைப்படுகிறது.

கணிப்பு வரிசை விதிகள்

(i) ஒவ்வொரு தொடரிலும் கணிப்பு இடமிருந்து வலமாகச் செல்லும்.

(ii) ஒரு தொடரிலுள்ள படிஉயர்த்தல் (Exponentiation) முதலில் செய்யப்படும்.

(iii) இரண்டாவதாகப் பெருக்கல், வகுத்தல் இரண்டுமே அவைகள் இடமிருந்து வலமாகப் போகும்போது எதிர்ப்படும் வரிசையில் செய்யப்படும்.

(iv) மூன்றாவதாகக் கூட்டல், கழித்தல் இரண்டும் அவை இடமிருந்து வலமாகப் போகும்போது எதிர்ப்படும் வரிசையில் செய்யப்படும்.

எடுத்துக்காட்டு

தொடர்: 17-35 * 4 + 4 * * 8 — 50 +
2 * * 3 / 2 + 46 * 2

மேற்கண்ட தொடரில் நாம் இடப் புறமிருந்து செல்லுகையில் முதன் முதலில் 4 * * 3-ஐயும் செய்துகொண்டு பின் 2 * * 3-ஐயும் செய்யவேண்டும். இதைச் செய்து முடித்த பிறகு மேற்கண்ட தொடர் 17—35 * 4 + 64 — 50 + 8 / 2 + 46 * 2 என்றாகிறது. அடுத்துப் பெருக்கல், வகுத்தல் இரண்டையுமே இடமிருந்து வலமாகச் செய்யவேண்டும். அப்படிச் செய்யும்போது 35 * 4 முதலிலும், 8/2 அடுத்தும், 46 * 2 அதற்கடுத்தும் செய்யப்படும். இரண்டையும் செய்து முடித்த பின்னர் நமக்குக் கிடைப்பது 17—140 + 64 — 50 + 4 + 92 ஆகும். அதற்குப் பின்னர் எஞ்சியுள்ள கூட்டல், கழித்தல் இரண்டையும் இடமிருந்து வலமாகச் செய்யவேண்டும். முதலில் (17—140)-ம் அத்துடன் 64-ஐக் கூட்டி அதிலிருந்து 50-ஐக் கழித்து அத்துடன் 4-ஐக் கூட்டி அத்துடன் 92-ஐக் கூட்ட வேண்டும். விடை 13 ஆகும்.

சில சமயங்கள் சாதாரணக் கணக்குகளில் பயன்படுத்துவதைப்போல அடைப்புக் குறிகளையும் நாம் ஃபோர்ட்ரானில் பயன்படுத்தலாம்.

உதாரணம்: 6 * * (4—2) / 5 + 16

ஒரு தொடரில் அடைப்புகள் பயன்படுத்தப்பட்டால் சற்று முன் கூறப்பட்ட கணிப்பு வரிசை முறைகளில் சற்று மாறுதல் ஏற்படும். அடைப்புக் குறிக்குள் இருப்பது **ஏதுவாயினும்** அதுவே இப்போது **முதலில்** கணிக்கப்படும். அடைப்புக் குறிக்குள் உள்ள தொடருக்கு மேற்கூறிய விதிகளைப் பயன்படுத்த வேண்டும்.

எடுத்துக்காட்டு 1.

6. * * (4. — 2.) / 5. + 16.

என்ற தொடரைக் கணிக்க, முதல் படி

6. * * 2. / 5. + 16. ஆகும்.

இரண்டாம் படியில்

36. / 5. + 16. கிடைக்கும்.

மூன்றும் படியில்

7. 2 + 16. கிடைக்க இறுதியில்

23. 2 என்ற விடை பெறப்படும்.

இதே தொடரை, அடைப்புகள் இல்லாமல் இருந்திருந்தால் அது 6. * * 4. — 2. / 5. + 16. என்றாகி, இறுதி விடையே வேறு விதமாக (1311. 6) இருக்கும் என்பதை அறிக.

ஒரு தொடரில் ஒன்றுக்கு மேற்பட்ட அடைப்புகள் இருப்பின் அவற்றுள் யாவற்றிற்கும் உள்ளிருக்கும் அடைப்புகளில் உள்ள தொடரே முதலில் கணிப்பிற்கு எடுத்துக் கொள்ளப்படுகிறது.

எடுத்துக்காட்டு 2.

4 * (.(3. — 5.) + 15.)

இதில் (3. — 5.) முதலில் சுருக்கப்பட்டு 4.* (— 2. + 15.) என்றாகி, பின்னர் 4. * 13. என்றாகி, பின்னர் இறுதி விடையான 52. கிடைக்கிறது.

எடுத்துக்காட்டு 3.

3. — (15. * 2. * * 2. / 3. + 4. * 2.)

இதைப் படிப்படியாகச் சுருக்க

3. — (15 * 4. / 3. + 4. * 2.)

3. — (20 + 8.)

3. — (20. + 8.)

3. — 28.

— 25.

என்று விடை கிடைக்கும்.

ஃபோர்ட்ரான் ஆணைகள் (Fortran Statements)

ஃபோர்ட்ரானின் ஆணைகள், அடிப்படையில் முக்கியமாக இரண்டு வகைப்படும். ஒன்று கணிப்பாணைகள்; மற்றது கொள்/கொடு ஆணைகள்.

கணிப்பாணையில் ஒரு ' = ' குறியும், இக் குறிக்கு இடப்புறம் ஒரு பெயர்ச்சொல்லும் வலப்புறம் ஒரு தொடரும் இருக்கும்.

உதாரணம் 1

BALAN = PAY — TAX + 150

இத்தகைய கணிப்பாணைகளில் வரும் ' = ' குறிக்கும், கணிதத்தில் பயன்படுத்தும் ' = ' குறிக்கும் வேறுபாடு உண்டு. கணிதத்தில் ' = ' குறிக்கு 'அளவில் சமம்' என்று பொருள். ஆனால் ஃபோர்ட்ரானில் ' = ' குறிக்கு வலப்புறத்தில் உள்ள தொடரைச் சுருக்கி வந்த விடையை ' = '-ன் இடப்புறம் உள்ள பெயரின் மதிப்பாகக் கொள் என்று பொருள்.

எடுத்துக்காட்டாக Pay-ன் மதிப்பு 1500 என்றிருந்து Tax-ன் மதிப்பு 175 ஆக இருப்பின், வலப் புறத்தைச் சுருக்க (மேலுள்ள உதாரணத்தில்) 1475 என்ற எண் கிடைக்கும். இந்த எண்ணே ' = '-க்கு இடப்புறம் உள்ள Balan என்ற சொல்லின் மதிப்பாகும். கணித முறைப்படி X = X + 15 என்ற ஒரு சமன்பாடு அர்த்தமற்றது. ஆனால் ஃபோர்ட்ரானில் இத்தகைய தொடர் பொருளுள்ளது மட்டுமன்றி, அடிக்கடி பயன்படுத்தப்படும் ஓர் ஆணையாகவும் உள்ளது.

எடுத்துக்காட்டாக X-ன் மதிப்பு 15 என்றால் இந்த மதிப்பை X = X + 20 என்று பயன்படுத்தும்போது வலப் புறத்தைச் சுருக்க, 15 + 20 = 35 ஆகிறது. இதை இடப்புறம் உள்ள பெயரின் மதிப்பாகக் கொள்ளும்போது X-ன் முந்தைய மதிப்பான 15 இழக்கப்பட்டு, X-ன் புதிய மதிப்பு 35 என்றாகிறது.

மாதிரி புரோகிராம்

இப்போது ஒரு மாதிரி புரோகிராம் ஒன்றைப் பார்ப்போம். X-ன் மதிப்பு —27.8 Y-ன் மதிப்பு 5.32; Z = X + Y-ஐப் பயன்படுத்தி Z-ன் மதிப்பைக் கணிக்க வேண்டும். அந்தப் புரோகிராமின் கட்டளைக் கொத்து கீழே கொடுக்கப்பட்டுள்ளது.

```
X =  — 27.8
Y =     5.32
Z = X + Y
PRINT,  Z
END
```

புரோகிராம் 5.1

மேலுள்ள ஐந்து வரிகளும் சேர்ந்து நமது Z-ன் கணிப்புக்கான ஒரு புரோகிராம் ஆகிறது. முதல் மூன்று வரிகளும் கணிப்பு ஆணைகளாம். நான்காவதாக உள்ள PRINT, Z என்பது ஒரு கொடு-ஆணை (output instruction or output statement) ஆகும்.

மேலுள்ள ஐந்து வரிகளையும் ஒரு கார்டில் ஒரு வரி என்ற முறையில் துளைத்து மேலுள்ள ஒழுங்கிலேயே அடுக்கிக்கொள்ளு

கிறோம். முதல் கார்டில் X = — 27.8 என்பதும், இரண்டாவதில் Y = 5.32 என்பதும், மூன்றுவதில் Z = X + Y என்ற கணிப்பாணையும், நான்காவதில் PRINT, Z என்ற கொடு-ஆணையும், முடிவாக ஐந்தாவது கார்டில் END என்ற வார்த்தையும் துளேக்கப்பட்டிருக்கும்.

இப்போது நமது முதல் வேலை இவ்வாணைகளை கம்ப்யூட்டரின் நினைவகத்தில் ஏற்றுவதாம். ஆனால் மேற்கண்ட ஆணைகளைக் கம்ப்யூட்டர் புரிந்துகொள்ள இயலாது. முதலில் அவற்றைக் கம்ப்யூட்டரின் பொறிமொழியில் மொழிபெயர்த்தாக வேண்டும். இந்த மொழிபெயர்ப்பைச் செய்யவேண்டியது ஃபோர்ட்ரான் கம்பைலர். ஃபோர்ட்ரான் கம்பைலர், கம்ப்யூட்டரின் நினைவில் முதலில் ஏற்படுகிறது. அதன் பின்னரே நமது புரோகிராம் கார்டுகள் கம்ப்யூட்டருக்கு அளிக்கப்படுகின்றன. இவ்வாறு அளிக்கப்படும்போது, முதல் ஆணையையுடைய கார்டு முதலிலும், இரண்டாவது ஆணையைக் கொண்டுள்ள கார்டு இரண்டாவதாகவும், இன்னும் இவ்வாறு மற்றக் கார்டுகளும் செலுத்தப்பட வேண்டும்.

கார்டுகள் ஒவ்வொன்ருக உள்ளே செல்லும்போது, கார்டில் உள்ள ஆணைகள் படிக்கப்பட்டு, கம்ப்யூட்டரின் நினைவகத்திற்கு அனுப்பப்படுகின்றன. அங்கே தயாராகக் காத்திருக்கும் ஃபோர்ட்ரான் கம்பைலர், வரும் ஃபோர்ட்ரான் ஆணையை, முதலில் அந்த ஆணை ஃபோர்ட்ரான் இலக்கணப்படி சரியானதா என்று சோதிக்கிறது. பிழை ஏதுமிருப்பின் உடனே, அந்தக் கார்டு கட்டில் எத்துணையாவது கார்டு என்பதையும், அதில் என்ன தவறு உள்ளது என்பதையும், கம்ப்யூட்டருடன் இணைக்கப்பட்டுள்ள டைப்ரைட்டரில் அச்சிடுகிறது. பிழையற்ற கார்டாக இருப்பின், அதிலுள்ள ஃபோர்ட்ரான் ஆணையைப் பொறிமொழி ஆணை அல்லது ஆணைகளாக மாற்றிக் கம்ப்யூட்டரின் நினைவகத்தின் ஓரிடத்தில் வைத்துக்கொள்ளுகிறது. ஒரு புரோகிராமில், ஒரு கார்டில் ஒரு சிறு தவறு இருப்பினும் (கம்ப்யூட்டருக்குச் சிறிய தவறு, பெரிய தவறு என்றெல்லாம் பாகுபாடு கிடையாது) அந்தப் புரோகிராம் கம்ப்யூட்டரால் நிராகரிக்கப்படும். குறையுள்ள கார்டின் பிழையைத் திருத்தி மீண்டும் நமது புரோகிராமைக் கம்ப்யூட்டருக்கு அளிக்கவேண்டும்.

மேலுள்ள புரோகிராமில் குறையேதுமில்லை எனக் கொள்ளுவோம். X = —27.8 என்ற கார்டில் ஆரம்பித்து ஒவ்வொரு கார்டாகப் படித்து அந்தக் கார்டில் உள்ள ஆணை, அதற்கிணையான பொறிமொழி ஆணைகளாக மாற்றப்பட்டுக் கம்ப்யூட்டரின்

நினைவகத்தில் ஒன்றன் பின் ஒன்றாக இருத்தி வைக்கப்படுகின்றன. இவ்வாணைகள் ஏதும் இப்போது செயல்படுத்தப்படுவதில்லை. அத்தியாயம் 1-ல் நமது சிறுவன் நம்மிடமிருந்து கட்டளைகளை ஒவ்வொன்றாகக் கேட்டுக்கொண்டு பிறகு செயல்படுகிறான் என்று பார்த்தோமல்லவா? அதில் அவன் நம்மிடமிருந்து கட்டளைகளைக் கேட்டுக்கொள்ளும் நிலையே, இந்நிலையாகும். சிறுவனுக்கு நாம் கட்டளைகளை ஒவ்வொன்றாகக் கூறிக்கொண்டே வந்து, கட்டளைகள் யாவற்றையும் அவனுக்கு அளித்த பிறகு, 'கட்டளைகள் இவ்வளவே, இனி உன் பணியைத் தொடரலாம்' என்றோ அல்லது அதற்கு ஈடான வார்த்தைகளையோ கூறி 'கட்டளையிடும் பணியை' முடித்துக் கட்டளைகளை நிறைவேற்றும் பணியை ஆரம்பித்து வைக்கிறோம். இதே முறையில் ஃபோர்ட்ரான் கம்பைலரும், ஒவ்வொரு ஃபோர்ட்ரான் ஆணையாக ஏற்று, மொழிபெயர்த்து நினைவகத்தில் சேர்த்துக்கொண்டே வரும்பொழுது, எல்லா ஆணைகளும் முடிந்தபிறகு 'இனி மொழிபெயர்ப்பதற்கு ஒன்றும் இல்லை, மொழிபெயர்க்கப்பட்ட ஆணைகளை நிறைவேற்றும் பணியைக் கம்ப்யூட்டர் ஆரம்பிக்கலாம்' என்று கூறியாகவேண்டும். இதையே END என்ற சொல் செய்கிறது. இவ்வாறு மொழிபெயர்ப்பு முடிந்ததும், கம்ப்யூட்டர் தானாக முதல் ஆணைக்குச் சென்று வரிசைக் கிரமமாக ஒவ்வோர் ஆணையாகச் செய்துகொண்டே வருகிறது. இவ்விடத்தில் இன்னொன்று கூறியாகவேண்டும்.

ஒரு கணிப்பாணையில் '=' குறிக்கு வலப்புறம் இருக்கும் ஓர் தொடரின் எண்களும், பெயர்களும் இருப்பின் பெயர் ஒவ்வொன்றிற்கான எண் மதிப்பு, ஒன்று வெளிப்படையாகக் கொடுக்கப்படல் வேண்டும்; அல்லது அந்தக் கணிப்பாணைக்கு வருவதற்கு முன்னர்க் கணித்துக் கண்டுபிடித்திருக்க வேண்டும். முன் பார்த்த புரோகிராமில் X, Y ஆகியவற்றின் மதிப்புகள் வெளிப்படையாகக் கொடுக்கப்பட்டுள்ளன.

இவ்விரண்டு எண் மதிப்புகளையும் கொண்டு X + Y கணிக்கப்பட — 22.48 கிடைக்கிறது. இந்த எண்தான் இப்போது Z-ன் மதிப்பாகும். அடுத்துவரும் PRINT, Z என்ற ஆணைப்படி, கம்ப்யூட்டருடன் தொடர்பு கொண்டுள்ள டைப்ரைட்டர் (அல்லது வேறு அச்சிடும் பொறி) வழியாக Z-ன் மதிப்பு அச்சிடப்படுகிறது.

மேலுள்ள புரோகிராமில் ஒரு சிறு மாறுதல் செய்வோம். முதலில் கொடுக்கப்பட்ட வரிசையை மாற்றி

```
X  =  —27.8
Y  =  5.32
  PRINT,  Z
Z  =  X + Y
    END
```

புரோகிராம் 5.2

என்ற வரிசையில் கார்டுகளை அடுக்கிக் கொடுப்பதாக வைத்துக் கொள்வோம். எந்த ஒரு புரோகிராமிலும் ஆணைகள், முதல் ஆணையில் ஆரம்பித்து வரிசைக் கிரமமாக ஒன்றன்பின் ஒன்ருக நிறைவேற்றப்படுகின்றன என்பதை நாம் ஏற்கெனவே அறிவோம்.

புரோகிராம் 5.2-ல் முதல் ஆணையிலிருந்து ஒவ்வொன்ருகச் செய்துகொண்டு வரும்போது மூன்ரும் ஆணைக்கு வந்ததும் PRINT, Z என்ற ஆணை எதிர்ப்படுகிறது. ஆனல் Z-ன் மதிப்பு இந்த இடத்தில் அறியாததொன்று. (அது இனிமேல்தான், அடுத்த வரிசையில், கணிக்கப்படும் என்பதை அறிக.) அதனல் கம்ப்யூட்டர் இந்த இடத்தில் தான் 'அறியாத' ஒரு பெயரின் மதிப்பை நாம் அச்சிடச் சொல்லுகிறேம்' என்ற பொருள்பட 'பிழைச் செய்தி' (error message) ஒன்றைக் கொடுக்கிறது. ஆக ஒரு புரோகிராமின் ஆணைகள் ஒரு தர்க்கரீதியான (logical) வரிசையில் இருப்பது அவசியம்.

புரோகிராம் 5.1-ல் X, Y ஆகிய இரண்டு சொற்களின் மதிப்பு புரோகிராமிலேயே கொடுக்கப்பட்டுள்ளன. அதை மாற்றி அந்தப் புரோகிராமைக் கீழ்க்கண்டவாறு எழுதலாம்.

```
READ  X, Y
Z  =  X + Y
PRINT,  Z
   END
—27.8,  5.32
```

புரோகிராம் 5.3

புரோகிராம் 5.3-ல், READ X, Y என்ற ஓர் ஆணை உள்ளது. இதற்கு 'X, Y-களின் மதிப்பை ஒரு கார்டிலிருந்து படித்துக்கொள்' என்று பொருள். மொழிபெயர்ப்பின்போது முதல் நான்கு (படம் 5.3) கார்டுகளே உள்ளே செல்லும் என்பதை நோக்குக. END கார்டு, மொழிபெயர்ப்பை முடித்து, கம்ப்யூட்டர் ஆணைகளைச் செயல்படுத்த ஆரம்பிக்கும்போது முதல் ஆணை READ, X, Y என்றிருக்கும். அதே சமயத்தில் —27.8, 5.32 என்ற எண்களைக்கொண்ட கார்டும் படிப்பிடத்தில் தயாராக இருக்கிறது. 'READ' ஆணை செயல்படும்போது இந்த இரண்டு எண்களைக்கொண்ட கார்டு படிக்கப்பட்டு X-ன் மதிப்பு —27.8 ஆகவும், Y-ன் மதிப்பு 5.32 ஆகவும் பெறப்படுகின்றன. புரோகிராம் 5.3-ல் —27.8, 5.32 என்ற எண்கள் டேட்டா (Data)—அதாவது அடிப்படை விவரம்—ஆகும். இந்த அடிப்படை விவரங்களைக்கொண்ட கார்டு 'டேட்டா கார்டு' ஆகும்.

சாதாரணமாக ஒரு புரோகிராமில் இரண்டு பகுதிகள் உண்டு. ஒன்று ஆணைப்பகுதி (Instruction Part) மற்றது 'டேட்டா' பகுதி; இந்த இரண்டாம் பகுதி சில புரோகிராம்களில் இல்லாமலும் போகலாம். எடுத்துக்காட்டு புரோகிராம் 5.1.

புரோகிராம், 5.1-ம் 5.3-ம் ஒரே கணிப்பைத்தான் செய்கின்றன என்றாலும் இரண்டின் அமைப்பிலும் வேறுபாடுகள் இருப்பதைக் காணலாம். புரோகிராம் 5.1-ல் X, Y ஆகியவற்றின் மதிப்புகள், புரோகிராமிலேயே கொடுக்கப்பட்டுள்ளன. இவற்றை மாற்றி, Z-ஐக் கணிக்க வேண்டுமாயின் நாம் புரோகிராம் கார்டுகளையே மாற்றவேண்டியிருக்கிறது. ஆனால் புரோகிராம் 5.3 அப்படியன்று. புரோகிராம் அப்படியேயிருக்க அடிப்படைக் கார்டுகளை (டேட்டா கார்டுகளை) மட்டும் மாற்றி X, Y களின் வெவ்வேறு மதிப்புகளுக்கு Z-ஐக் கணிக்க முடியும். (சாதாரணமாகப் புரோகிராம் எழுதும்போது இந்த முறையைத் தான் பின்பற்றவேண்டும். அப்போதுதான் அது ஒரு பொதுப் படைப் புரோகிராமாக (General Program) இயங்க முடியும்.

மேற்கண்ட புரோகிராம் 5.3-ல் ஒரேஒரு டேட்டா கார்டு கொடுக்கப்பட்டுள்ளது. இப்போது பலவேறு ஜோடி X, Y மதிப்புகளுக்கு Z-ன் மதிப்பைக் கணித்து அதை அச்சிடவேண்டியிருப்பதாகக் கொள்ளுவோம். ஒவ்வொரு கார்டிலும் இரண்டு எண்கள் (முதல் எண் X-ன் மதிப்பு இரண்டாவது எண் Y-ன் மதிப்பு) துளைக்கப்பட்டு ஒரு கற்றை டேட்டா கார்டுகள் இருப்பதாகக் கொள்ளுவோம். இந்தக் கற்றையிலுள்ள ஒவ்வொரு கார்டின் X, Y மதிப்புகளுக்கும் Z-ஐக் கணித்து மதிப்பிடவேண்டு

மாயின் புரோகிராம் 5.3-ஐ ஒவ்வொருமுறை கம்ப்யூட்டரில் ஓட்டவேண்டியுள்ளது. இது மிகவும் அசௌகரியமானதுமட்டுமல் லாமல், வீண் காலவிரயமும் ஆகிறது. ஒரு புரோகிராமை நினை வகத்தில் இட்ட பின்னர் அந்தப் புரோகிராமின் ஆணைகள் கம்ப் யூட்டரின் நினைவகத்தில் அழியாதிருக்குமாதலால், புரோகிராமைத் திருப்பித் திருப்பி ஓட்டாமல், முதலில் கொடுத்த புரோகிராமையே வைத்துக்கொண்டு டேட்டா கார்டுகளை மட்டும் ஒவ்வொன்றுகப் படித்து ஒவ்வொன்றிற்கும் உரிய Z ஐக் கணித்து அச்சிட ஓர் ஏதுவிருப்பின் அது மிகவும் சௌகரியமாக இருக்கும். ஃபோர்ட் ரானில் அத்தகைய வசதியுமுள்ளது. புரோகிராம் 5.4ஐக் காண்க.

```
10 READ X, Y
Z = X + Y
PRINT, Z
GO TO 10
END
```

} புரோகிராம் பகுதி

```
—27.8,   5.32
65.2,   —3.67
47.3,    4.325
92.7,   43.72
```

} டேட்டா பகுதி

புரோகிராம் 5.4

புரோகிராம் 5.3-க்கும், 5.4-க்கும் இரண்டு வேறுபாடுகள் உண்டு. புரோகிராம் 5.3-ல் இல்லாத இரண்டு அம்சங்கள் புரோகிராம் 5.4-ல் உள்ளன. ஒன்று READ X, Y என்ற ஆணை யுடன் காணப்படும் 10 என்ற எண். மற்றது புதிதாகக் காணப் படும் GO TO 10 என்ற ஆணை. ஃபோர்ட்ரானின் ஒவ்வோர் ஆணைக்கும், தேவை ஏற்படின் நாம் ஓர் ஆணை-எண் (Statement Number) கொடுக்க வசதி இருக்கிறது.

இப்போது புரோகிராம் 5.4 எவ்வாறு செயல்படுகிறது என்று சற்றுப் பார்ப்போம். புரோகிராமின் ஆணைகள் செயல்படுத்தப்பட

ஆரம்பிக்கும்போது, முதல் ஆணை 10 READ X, Y என்றிருக்கும். அப்போது டேட்டா பகுதியைச் சேர்ந்த கார்டுகளில், முதல் கார்டான —27.8, 5.32 கார்டு படிக்கப்படக் காத்திருக்கிறது. அதனால் 'READ' ஆணை செயல்பட ஆரம்பித்ததும் இந்தக் கார்டு படிக்கப்பட்டு X-ன் மதிப்பு —27.8 என்றும், Y-ன் மதிப்பு 5.32 என்றும் ஆகிறது. (மற்ற டேட்டா கார்டுகள் இப்போது படிக்கப்படமாட்டா.) இந்த X, Y மதிப்புகளைக்கொண்டு அடுத்த ஆணையான Z = X + Y, Z-ன் மதிப்பைக் கணக்கிடுகிறது. அடுத்த ஆணை (PRINT, Z) Z-ன் மதிப்பை அச்சிடச் செய்கிறது. இந்த ஆணை முடிந்தவுடன்

GO TO 10

என்ற ஆணை எதிர்ப்படுகிறது. இந்த ஆணை, கம்ப்யூட்டருக்கு 'ஆணை-எண் 10 கொண்ட ஆணைக்கு உடனே பிரிந்து சென்று அங்குக் கூறப்பட்டுள்ளதைச் செய்' என்று ஆணையிடுகிறது. இதன்படி கம்ப்யூட்டர் மீண்டும்

10 READ X, Y

என்ற ஆணைக்குச் சென்று அதைச் செயல்படுத்துகிறது. இந்த ஆணை இப்போது இரண்டாம் முறையாகச் செயல்படுகிறது என்பதை அறிக. இவ்விரண்டாம் முறைச் செயல்படும்போது முதல் டேட்டா கார்டு ஏற்கெனவே படிப்பிடத்தை விட்டுப் போய்விட்டிருக்கும். இரண்டாவது டேட்டா கார்டு

65.2, —3.67

படிப்பிடத்தில் காத்திருக்கும். இரண்டாம் முறை 'READ' ஆணைச் செயல்படும்போது இந்தக் கார்டு படிக்கப்பட்டு X-ன் மதிப்பு 65.2 ஆகவும் Y-ன் மதிப்பு —3.67 ஆகவும் ஆகிறது.

இந்தக் கார்டைப் படித்தவுடன் கம்ப்யூட்டர் புரோகிராமில் அதற்கடுத்த ஆணையான 'Z = X + Y'க்குத் தானாகவே மீண்டும் செல்லுகிறது. அங்கே Z-ஐக் கணித்த பின்னர் அடுத்த ஆணையான PRINT, Z என்ற ஆணையை இரண்டாம் முறையாக நிறைவேற்றுகிறது. இப்போது Z-ன் புதிய மதிப்பு அச்சிடப்படுகிறது. மீண்டும் GO TO 10 என்ற ஆணை, கம்ப்யூட்டரை 10 READ X, Y என்ற ஆணைக்குத் திருப்ப, மூன்றாவது டேட்டா கார்டு படிக்கப்பெற்று மற்ற இரு கார்டுகளைப்போலவே பயன்படுத்தப்படுகிறது. இவ்வாறாக ஒவ்வொரு கார்டையும் படித்து அதிலுள்ள X, Y மதிப்புகளைக்கொண்டு Z-ஐக் கணித்து அதை அச்சிட்ட பின்னர், மீண்டும் ஒரு கார்டைப் படிக்கக் கம்ப்யூட்டர் தயாராயிருக்கிறது. இந்தப் புரோகிராம்படி டேட்டா கார்டுக் கற்றையில் எவ்வளவு கார்டுகள் இருந்தாலும் அவற்றை

எல்லாவற்றையும் முடித்த பிறகு மீண்டும் இன்னொரு கார்டைப் படிக்கக் கம்ப்யூட்டர் காத்திருக்கும்.

GO TO

என்ற ஃபோர்ட்ரானின் ஆணை வசதிமிக்க ஓர் ஆணை யாகும். இதன் சௌகரியத்தின் ஒரு சிறு எடுத்துக்காட்டே நாம் மேற் கண்ட பத்திகளில் பார்த்தவை. GO TO ஆணைதவிர இன்னும் சில முக்கியமான ஃபோர்ட்ரான் ஆணைகளும் உள்ளன. அவற் றுள் கட்டளைப் பிரிவு (Test Statement) 'IF' ஆணை முக்கிய மானது. இந்த ஆணையின் மாதிரி :

IF (Y) 15, 20, 30

மேற்கண்ட 'IF' ஆணையின் விளக்கம்: 'Y-ன் மதிப்பு —ve ஆனல், புரோகிராமில் ஆணை எண் 15 உள்ள இடத்திற்குச் சென்று அங்குக் கூறப்பட்டதைச் செய்; Y-ன் மதிப்பு 0-ஆனல் ஆணை-எண் 20-க்குச் சென்று அங்குக் கூறப்பட்டதைச் செய்; Y-ன் மதிப்பு + ve ஆனல் ஆணை-எண் 30-க்குச் சென்று அங்குக் கூறப்பட்டுள்ளதைச் செய்.' இந்த ஆணையின் பயன் எளிதில் விளங்கும். இதன் முக்கியத்துவமே, இதன் ஒப்பு நோக்கி முடி வெடுத்து புரோகிராமின் மூன்று பாகங்களுக்குப் பிரிந்து செல் லும் ஆற்றலே. இந்த ஆணையின் பயனை அறிந்துகொள்ள மேலும் ஒரு சிறு உதாரணத்தைக் காண்போம்.

ஒரு கற்றைக் கார்டுகள் கொடுக்கப்பட்டுள்ளன. இந்தக் கற்றையிலுள்ள கார்டு ஒவ்வொன்றிலும் ஓர் எண் துளைக்கப் பட்டிருக்கிறது. இந்த எண்கள் X-ன் மதிப்புகளாகும். ஒரு கார்டில் உள்ள எண்ணைப் படித்ததும், அது '—' குறியுடைய எண்ணனல் (Negative Number) $Y = X^2$ என்ற சூத்திரத்தை வைத்து Y-ஐக் கணித்து அச்சிடவேண்டும்; 0 ஆனல் ஒன்றும் செய்யாமல் அந்தக் கார்டைப் புறக்கணித்துவிட்டு, அடுத்த கார்டுக்குச் செல்லவேண்டும்; இவ்விரண்டுமில்லாத எண்ணாயின் (+ ve Number) Y-ன் மதிப்பை $Y = -2X$ என்ற சூத்திரத் தைப் பயன்படுத்திக் கணித்து அச்சிடவேண்டும்.

இப்போது நமக்கு மேற்கண்ட பாராவில் கூறியவற்றைச் செய்யவல்ல புரோகிராம் ஒன்று தேவை.

நமது புரோகிராம் என்னென்ன செய்யவேண்டும் என்று பார்ப்போம்.

(i) கார்டில் உள்ள எண்ணைப் படிக்கவேண்டும்.

(ii) படித்த எண்ணைச் சோதிக்கவேண்டும். எண் — குறியானுல் iii-ல் உள்ள கட்டளையையும், 0 ஆனுல் i-ல் உள்ள கட்டளையையும், இரண்டுமில்லையாயின் iv-ல் உள்ள கட்டளையையும் செய்யவேண்டும்.

(iii) Y-ன் மதிப்பை $Y = X^2$-ஐக் கொண்டு கணித்து அச்சிட்ட பின்னர் i-க்குச் செல்லவேண்டும்.

(iv) Y-ன் மதிப்பை $Y = -2X$-ஐக் கொண்டு கணித்து அச்சிட்ட பின்னர் நிறுத்திக்கொள்.

இப்போது (i) (iii) (iv) ஆகியவற்றிற்கான ஃபோர்ட்ரானின் ஆணைகளை எழுதுவோம்.

(i)

```
15 READ, X
```

(ii) IF (X) 30, 15, 20

(iii)

```
30 Y = X * X
PRINT, Y
GO TO 15
```

(iv)

```
20 Y = —2. * X
PRINT, Y
END
```

மேற்கண்ட மூன்று சிறு பகுதிகளையும் ஒன்றுக இணைத்து ஒரு புரோகிராம் தயார் செய்யவேண்டும். இதைச் செய்வது எவ்வாறு?

முதலில் Y = X * X என்ற ஆணைக்கு ஆணை-எண் 30-ம் Y = —2. * Xக்கு ஆணை-எண் 20-ம் கொடுப்போம்.

ஒரு கார்டில் X-ன் மதிப்பைப் படித்த வுடனே அதன் மதிப்பு — ve, o, அல்லது + ve இவற்றுள் எது என்று காணல் வேண்டும். அவ்வாறு கண்டபிறகு — ve ஆக இருப்பின், ஆணை எண் 30-க்கும், o-ஆக இருப்பின் ஆணை எண் 15-க்கும் + ve ஆக இருப்பின் ஆணை-எண் 20-க்கும் செல்ல வேண்டும். இதற்கான 'IF' ஆணை.

IF (X) 30, 15, 20 ஆகும். இவ்வாணையை,

15 READ, X

ஆணைக்கு அடுத்தபடியாக வைத்து முடிவில் END — ஆணை யையும் சேர்க்கக் கீழ்க்கண்டவாறு நமது புரோகிராம் முடிவு பெறுகிறது.

```
15  READ,  X
    IF (X) 30, 15, 20
30  Y = X * Y
    PRINT,  Y
    GO  TO  15
20 Y = —2. * X
    PRINT, Y
    END
```

புரோகிராம் 5.5

இதுவரை நாம் பார்த்தவை ஃபோர்ட்ரானின் ஒரு மிகச் சிறு பகுதியே ஆயினும் ஒரு கம்ப்யூட்டர் புரோகிராம் எப்படி வேலை செய்கிறது என்பதை இது ஓரளவு எடுத்துக் காட்டுகிறது. புரோகிராம் செய்யப் பயன்படுத்தும் மொழி எதுவாயினும், கம்ப் யூட்டருக்கான ஆணைகளின் தன்மை நாம் சற்று முன் பார்த்த புரோகிராம்களைப் போலத்தான் இருக்கும். கணிப்பு முறை சிக்க லுடையதாக ஆகும்போது புரோகிராமும் பெரிதாகவும், சிக்கல் வாய்ந்ததாகவும் ஆகும்.

ஃபோர்ட்ரான் மொழியில் எழுதிப் பயன் படுத்தப்பட்ட புரோகிராம் ஒன்று மாதிரிக்காகச் சேர்க்கப்பட்டிருக்கிறது. இது ஒரு குறிப்பிட்ட தேர்வு எழுதிய மாணவர்களது மதிப்பெண் களைக் கொண்டு, தேர்வு முடிவுகளைத் தயாரிக்கும் புரோகிராம் ஆகும்.

```
ZZFORX
* LDISKPOLY2
* FANDK08C6
C  C   POLY EXAMINATION DATA PROCESSING
C      NPAP (I) = PAPER NUMBERS
C      NSUB (I) = NUMBER OF SUBJECTS IN EACH
C      NP = TOTAL NUMBER OF PAPERS
C      NS = TOTAL NUMBER OF SUBJECTS
C      TOTMOD = TOTAL MARKS AVAILABLE FOR
                                        MODERATION
C      M (I) = MARKS IN EACH PAPER
C      MCOMB (I, J) = EXAM MARK (M(I) ) +
                                   SESSIONAL MARKS
C      MSESS (I, J) = SESSIONAL MARKS
C      M45 (I, J) 45 PERCENT
C      M40 (I, J) = 40 PERCET OF EACH PAPER
C      M35 (I, J) = 35 PERCET OF EACH PAPER
C      MCOND (I, J) MAXIMUM CONDONATION
C      MDEF (I, J) = DEFICIENCY IN ITH
                    SUBJECT AND JTH PAPER
C      MAX (I, J) = MAXIMUM MARKS IN SUBUECT
C      MINCO (I, J) = MINIMUM COMB MARKS
C      MFIRST = MINIMUM TOTAL FOR FIRST CLASS
C      MSECON = MINIMUM TOTAL FOR SECOND
                                             CLASS
C      LDEFS (I), NDEFS (I) = DEFICIENCY
                                     SUBJECT - WAR
C      NDEFP (I, J) = DEFICIENCY PAPER - WAR
C      TOT DEF = TOTAL DEFICIENCY
C      NSFAIL = NUMBER OF SUBJECTS FAILED
C      MEXEM (I, J) = EXEMPTION MARKS
       DIMENSION MAXDS (10), MAXDP (10, 2)
       DIMESION NPAP (11), NSUB (11), MCOMB
                          (11, 2), MSESS (11, 2),
       5M45 (11, 2), M40 (11, 2)
       DIMENSION MCDEF (10, 2) MCTDE (10)
       DIMENSION MCOND (11, 2), MDEF (11, 2),
                                      LDEFS (11),
       4NDEFS (11)
       DIMENSION        NDEFP (11, 2), M (11, 2)
```

```
      DIMENSION NDEP (11, 2), MEXEM (11, 2),
                    A (20), COMENT (4, 3), M35 (11, 2)
      DIMENSION M1 (11, 2) NCOND (11), MARK (11,2)
C     HEADER PUNCHING
      READ 1000, (A (I), I = 1, 20)
      PUNCH 1000, (A (I), I = 1, 20)
C
      READ 1010, ( ( COMENT (I, J), J = 1,3), 1 = 1,4)
      READ 1020, NS, NP, NTOTMO, NEXEM
      READ 1021, MFIRST
      READ 1015, (NSUB (I), NPAP (I), 1 = 1, NS)
      READ 1020, ( ( MCOND (I, J), J = 1, 2), 1 =
                    1, 4), MCOND (5, 1), MCOND (6, 1)
      READ 1020, ( ( M45 (I, J), J = 1, 2), 1 = 1, 4),
                                   M45 (5, 1), M45 (6, 1)
      READ 1020, ( ( M40 (I, J), J = 1, 2), I = 1, 4), )
                                  M40 (5, 1), M 40 (6, 1 )
      READ 1020, ( ( M35 (I, J), J= 1 = 2), 1= 1, 4),
                                   M35 (5, 1) M35 (6, 1)
      READ 1020, (( MEXEM (1, J), = 1, 2), 1 = 4),
                              MEXEM (5, 1), MEXEM (6, 1)
C     PROCESSING PART
 9999 READ 1030, NN, ( ( MSESS ( I, J), J = 1, 2),
                 1 = 1, 4), MSESS 5, 1), MSESS (6, 1),
      I ( ( M ( I, J ), J = 1, 2), 1 = 1, 4), M ( 5, 1 ), M ( 6, 1)
      I AB = - 1
 1000 FORMAT ( 20A4 )
 1010 FORMAT ( 20A4 )
 1015 FORMAT ( 8011 )
 1020 FORMAT ( 2613 )
 1021 FORMAT ( 5I4 )
 1030 FORMAT ( I6, 34X, 20I2 )
 5000 FORMAT ( I5, IX, 3A4, 10I2, 1X, 10I3, 1X, I5, 2X,
                                        1HE, 1HT, 1HB)
 6000 FORMAT ( I5, IX, 3A4, 10I2, 1 X, 10I3, 1X, 5X,
                                    2X, IHE, 1HT, 1HB)
      LEXEM=NEXEM
      LTOTMO=NTOTMO
      MTOTMO=LTOTMO
```

```
C      COMBINED TOTALS MODERATION FOR PASS
                                        CASES
C
C
       JAIL = - 1
       DO 3001 1 = 1, NS
       NPAP I = NPAP ( I )
       DO 3001 J = 1, NPAP I
       IF ( M ( I, J ) ) 978, 979, 979
  978  CONTINUE
       IAB = IAB + 1
  979  CONT INUE
       MCOMB ( I, J ) = MSESS ( 1, J ) + M ( I, J )
       MCDEF ( I, J ) = 0
       M 1 ( I, J ) = M ( I, J )
       I F ( M ( I, J ) - M35 ( I, J ) ) 3003, 3002, 3002
 3003 JAIL = JAIL + 1
       GO TO 3001
 3002 I F ( MCOMB ( I, J ) - M45 ( I, J ) ) 3004, 3001, 3001
 3004 MCDEF ( I, J ) = M45 ( I, J ) - MCOMB ( I, J )
       I F ( MEXEM ( I, J ) ) 4000, 3001, 4000
 4000 CONT INUE
       I F ( M ( I, J ) - M40 ( I, J ) ) 3005, 3006, 3006
 3005 CONTINUE
       I F (MCDEF ( I, J ) - MCOND ( I, J ) ) 3007, 3003,
 3007 MTOTMO = MTOTMO - MCDEF ( I, J )
       LEXEM = LEXEM - MCDEF ( I, J )
 3006 M ( I, J ) = M1 ( I, J ) + MCDF ( I, J )
       M1 ( I, J ) = M1 ( I, J ) + MCDEF ( I, J )
       MCOMB ( J, J ) = M ( I, J ) + MSESS ( I, J )
       MCDEF ( I, J ) = 0
 3001 CONTINUE
C
C      CLEAR PASS CASES
C
       MAG = 0
       I F ( JAIL ) 3010, 3311, 3011
 3010 DO 1 I = 1, NS
       NPAPI = NPAP ( I )
       DO 1 J = 1, NPAPI
```

```
1    MAG = MAG + MCOMB ( I , J )
     I F ( MAG - MF IRST ) 7, 6, 6
7    CONT INUE
     MAD = MF IRST - MAG
     I F ( MAD - 5 ) 790, 790, 791
790  M ( 3, 1 ) = M ( 3, 1 ) + MAD
     I I = 1
     GO TO 8
791  CONT INUE
     1 I = 2
     GO TO 8
6    1 1 = 1
8    CONT INUE
     PUNCH 5,000, NN, ( COMENT ( I I, J ), J = 1, 3 ),
                    ( ( M ( 1, J ) , J = 1 , 2 ), 1 = 1, 4 ) ,
     4M ( 5, 1 ) , M ( 6, 1 ) , ( ( MCOMB ( 1, J ) , J=1, 2 ) ,
          1=1, 4 ) , MCOMB ( 5, 1 ), MCOMB ( 6, 1 ) ,
     5 MAG
     GO TO 9999
3011 CONTINUE
C    COMPUTATION OF DEFICIENCIES IN EXAMINA-
                                  TION MARKS ONLY
C
C
C
C    COMPUTATION OF DEFICIENCIES
     NSFAIL = 0
     DO 20 I = 1, NS
     NDEFS ( I ) = 0
     NPAP I = NPAP ( I )
     DO 22 J = 1, NPAPI
     NDEFP ( I, J ) = 0
     IF ( M ( I, J ) - M35 ( I, J ) ) 21, 22, 22
21   NDEFP ( I, J ) = M35 ( I, J ) - M ( 1, J )
     NDEFS ( I ) = NDEFS ( I ) + NDEFP ( I, J )
22   COTINUE
     LDEFS ( I ) = NDEFS ( I )
     NSFAIL = NSFAIL + 1
20   CONTINUE
C
C    DEFICIENCIY IN COMBINED TOTALS
```

```
C       FOR FAIL AND ABSENT CASES
C
        DO 30 I = 1, NS
        MCTDE ( I ) = 0
        NPAPI = NPAP ( I )
        DO 30 J = 1, NPAPI
   31   IF (MCOMB ( I, J ) - M45 ( I, J ) ) 32, 30, 30
   32   MCDEF ( I, J ) = M45 ( I, J ) - MCOMB ( I, J )
        MCTDE ( I ) = MCTDE ( I ) + MCDEF ( I, J )
   30   CONTINUE
C
C       MODERATION
C
C       ASSEBLING HIGHER OF THE TWO DEFICIENCIES
C
        DO 33 I = 1, NS
        MAXDS ( I ) = 0
        NPAPI = NPAP ( I )
        DO 33 J = 1, NPAPI
        IF ( MCDEF ( I, J ) - NDEFP ( I, J ) ) 34, 35, 35
   34   MAXDP ( I, J ) = NDEFP ( I, J )
        GO TO 36
   35   MAXDP ( I, J ) = MCDEF ( I, J )
   36   MAXDS ( I ) = MAXDS ( I ) + MAXDP ( I, J )
   33   CONTINUE
C
C       EFFECTING MODERATION
C       IFAIL = NO OF FAILED SUBJECTS (INCLUDING
                                      WANT OF SESS, MARKS)
C
        IFAIL = 0
        DO 37 I = 1, NS
        IF ( MAXDS ( I ) ) 38, 37, 38
   38   IFAIL = IFAIL + 1
   37   CONTINUE
        DO 82 I K = I, IFAIL
        NSMALL = 1000
        DO  990  I = 1, NS
        IF ( MAXDS ( I ) ) 991, 990, 991
  991   IF ( MAXDS ( I ) - NSMALL ) 992, 990, 990
  992   NSMALL = MAXDS ( I )
        L = I
  990   CONTINUE
C
C
        IF ( MAXDS ( L ) - MTOTMO ) 805, 805, 906
  805   MX = 0
```

```
      NPAP L = NPAP ( L )
      DO, 807 J = 1, NPAPL
  807 MX = MX + MCOND ( L, J )
      IF ( MAXDS ( L ) - MX ) 808, 808, 82
  808 NPAPL = NPAP ( L )
      IF ( NPAP L - 1 ) 803, 803, 802
  803 M ( L, 1 ) = M ( L, 1 ) + MAXDP ( L, 1 )
      MTOTMO = MTOTMO - MAXDP ( L, 1 )
      MAXDS ( L ) = 0
      GO TO 82
  802 CONT INUE
      IF ( MAXDP ( L, 1 ) - MAXDP ( L, 2 ) ) 821, 821, 820
  821 IF ( MAXDP ( L, 1 ) ) 831, 830, 831
  831 IF ( MAXDP ( L, 1 ) - MCOND ( L, 1, ) ) 875, 875, 850
  875 M ( L, I ) = M ( L, 1) + MAXDP ( L, 1 )
      MTOTMO = MTOTMO - MAXDP ( L, 1 )
      K = 2
  605 IF ( MAXDP ( L, K ) - MTOTMO ) 600, 600, 601
  600 IF ( MAXDP ( L, K ) - MCOND ( L, K ) ) 602, 602, 601
  602 M ( L, K ) = M ( L, K ) + MAXDP ( L, K )
      MTOTMO = MTOTMO - MAXDP (L, K )
  601 CONTINUE
      MAXDS ( L ) = 0
      GO TO 850
  830 IF ( MAXDP (L, 2 ) - MCOND ( L, 2 ) ) 876, 876, 850
  876 M ( L, 2 ) = M ( L, 2 ) + MAXDP ( L, 2 )
      MTOTMO = MTOTMO - MAXDP ( L, 2 )
      K = 1
      GO TO 605
  820 IF ( MAXDP (L, 2 ) ) 830, 831, 830
  850 CONT INUE
  82  CONT INUE
      GO TO 806
C
C
C
  906 CONT INUE
      IF ( NPAP ( L ) - 1 ) 806, 806, 907
  907 DO 908 J = 1, 2
      IF ( MAXDP ( L, J ) ) 908, 806, 908
  908 CONT INUE
      IF ( MAXDP ( L, 1 ) - MAXDP ( L, 2 ) ) 909, 909, 910
  909 J = 1
  912 IF ( MAXDP ( L, J ) - MTOTMO ) 911, 911, 806
  911 IF ( MAXDP ( L, J ) - MCOND ( L, J ) ) 913, 913, 806
```

```
  913 M ( L, J ) = M (L, J ) + MAXDP ( L, J )
      MTOTMO = MTOTMO - MAXDP ( L, J )
      GO TO 806
  910 J = ?
      GO TO 912
C
  806 CONT INUE
C     CHECK FOR PASS AFTER MODERATION
C
      NAIL = - 1
      DO 851 I = 1, NS
      NPAPI = NPAP ( I )
      DO 853 J = 1, NPAP I
      IF ( M ( I, J ) - M 35 ( I, J ) ) 852, 854, 854
  852 NAIL = NAIL + 1
      GO TO 851
  854 IF ( M ( I, J ) + MSESS ( I, J ) - M 45 ( I, J ) ) 852,
                                                      853, 853
  853 CONT INUE
  851 CONT INUE
      IF ( NAIL ) 856, 857, 858
  856 I I = 2
      GO TO 859
  857 CONT INUE
      IF ( IAB ) 970, 971, 971
  970 CONT INUE
      I I = 3
  859 CONT INUE
      MAG = 0
      DO 860 I = 1, NS
      NPAP I = NPAP ( I )
      DO 860 J = 1, NPAP I
      MCOMB ( I, J ) = M ( I, J ) + MSESS ( I, J )
      MAG = MAG + MCOMB ( I, J )
  860 CONT INUE
      IF ( I I - 3 ) 9907, 8, 8
 9907 CONT INUE
      MMM = NTOTMO - MTOTMO
      IF ( MMM - 5 ) 792, 792, 793
  792 IF ( MAG - MF IRST ) 793, 794, 794
  794 I I = 1
      GO TO 8
  793 CONT INUE
      GO TO 8
C     SUBJECT EXEMPTION
C
  858 CONT INUE
```

```
      NAIL = JAIL + 1
      DO 945 KK = 1, NAIL
      NSMALL = 1000
      DO 900 I = 1, NS
      NPAP I = NPAP ( I )
      DO 900 J = 1, NPAP I
      IF ( MAX DP ( I, J ) ) 905, 900, 905
  905 IF ( MAX DP ( I, J ) - NSMALL ) 901, 900, 900
  901 NSMALL = MAXDP ( I, J )
      L 1 = I
      L 2 = J
  900 CONT INUE
C
C
      IF ( MAXDP ( L 1, L 2 ) - LEXEM ) 902, 902, 946
  902 CONT  INUE
      IF ( M 1 ( L 1, L 2 ) - 30 ) 946, 904, 904
  904 CONT INUE
      IF ( MEXEM ( L 1, L 2 ) ) 980, 946, 980
  980 CONT INUE
      M 1 ( L 1, L 2 )=M1 ( L 1, L 2 )+MAXDP ( L 1, L 2 )
      M ( L 1, L 2 )=M ( L 1, L 2 )+MAXDP ( L 1, L 2 )
      LEXEM = LEXEM - MAXDP ( L 1, L 2 )
  946 MAXDP ( L 1, L 2 ) = 0
  945 CONT INUE
  950 CONT INUE
C
  955 CONT INUE
      DO 951 I = 1, NS
      NPAP I = NPAP ( I )
      DO 951 J = 1, NPAP I
      M ( I . J ) = M 1 ( I, J )
  951 MCOMB ( I, J ) = M ( I, J ) + MSESS ( I, J )
  971 CONT INUE
      I I = 4
      PUNCH 6000, NN, ( COMENT ( I I, J ). J = 1, 3 ),
                      ( ( M ( I, J ), J = 1, 2 ), 1 = I, 4 ),
      4 M ( 5, 1 ), M ( 6, 1 ), ( ( MCOMB ( I, J ), J = 1, 2 ),
               I = 1, 4 ), MCOMB ( 5, 1 ), MCOMB ( 6, 1 )
      GO TO 9999
 3333 FORMAT ( 10I4 )
      END
```

6

கம்ப்யூட்டரின் பயன்கள்

கம்ப்யூட்டரால் என்ன பயன்? அதை எந்தெந்தத் துறைகளில் பயன்படுத்தலாம்? என்பன போன்ற கேள்விகளுக்கு விடையிறுப்பது அவ்வளவு எளிதன்று. சுருக்கமாகக் கூறவேண்டுமானல் கம்ப்யூட்டரை எந்தத் துறையில் வேண்டுமானலும் பயன்படுத்தலாம். மனித வாழ்க்கையில் 600-க்கும் மேற்பட்ட துறைகளில் கம்ப்யூட்டர் பயன்படுத்தப் படுகிறதென்றும், நாள் தோறும் இந்த எண்ணிக்கை அதிகரித்துக் கொண்டே போகிறது என்றும் கூறப்படுகிறது. எந்தத் துறையாயிருந்தாலும் எத் தகைய கணிப்பாயிருப்பினும், அக்கணிப்பு முறையை அல்லது இயக்கமுறையை, சூத்திர வடிவில் வடிக்க முடிந்தால் அந்தத் துறையில், அந்தக் கணிப்பைக் கம்ப்யூட்டர் செய்ய இயலும்.

விவசாய, உணவுத் துறையில் கம்ப்யூட்டர்

விவசாய ஆராய்ச்சியில் ஆரம்பித்து, பொருள்களின் உற்பத்தி, சுத்திகரிப்பு, வினியோகம் ஆகிய எல்லாத் துறைகளிலும் இன்று பயன்படுத்தப்படுகிறது. கேக் போன்ற உணவுப் பண்டங்கள் தயாரிப்பதில் எந்தெந்தப் பொருளை எந்தெந்த விகிதத்தில் கலந்தால், விலை குறைந்து சத்தும் கூடி சுவையுடன் தயாரிக்க முடியும் என்பதையும் இன்று கம்ப்யூட்டர் கணக்கிட்டுச் சொல்லுகிறது. அதுவுமல்லாமல் உணவுப் பொருள்களின் விலை மாறிக்கொண்டே இருப்பதால் ஒரு மனிதனுக்குத் தேவையான சத்துப் பொருள் குறையாமல் மிகக் குறைந்த செலவில் என்னென்ன பொருள்களை அவன் சாப்பிடலாம் என்ற செய்தி, மருத்துவ நிலையங்கள், அனதை விடுதிகள் போன்ற இடங்களுக்குத் தேவைப்படுகிறது. ஒரு மனிதன் இத்தகைய கணிப்பு ஒன்றைச் செய்யக் குறைந்தது 4-மணி நேரமாகும்; ஆனல் ஒரு கம்ப்யூட்டருக்குச் சில வினடிகள் போதுமானது.

மாட்டுப்பண்ணை, கோழிப்பண்ணை போன்றவற்றிலுங்கூட கம்ப்யூட்டர் எதிர்காலத்தில் நுழையப் போகிறது. ஒவ்வொரு பிராணியின் உடலிலும் ஓர் எலக்ட்ரானிக் சாதனம் ஒன்றைப் பொருத்துவார்கள். அச்சாதனம், அப்பிராணியின் உடல் நிலையின் பலவேறு அம்சங்களையும், ஏதாவது நோய் ஏற்படின் அதைப்பற்றிய விவரத்தையும் ஒரு கம்ப்யூட்டருக்கு அனுப்பிக் கொண்டேயிருக்கும். பிராணிகளின் உடல் நிலையைக் கண்காணிக்கும் இந்தக் கம்ப்யூட்டர், அப்போதைக்கப்போது பிராணிகளைப் பற்றிய பலவேறு செய்திகளையும் ஓர் அறிக்கையாக அச்சடித்துக் கொடுப்பதோடல்லாமல் எந்தெந்தப் பிராணி நோய்வாய்ப்பட்டிருக்கிறது; அதற்கு என்னென்ன சிகிச்சை அளிக்கப்பட வேண்டுமென்பதையும் பண்ணையாளருக்கு அறிவிக்கும். இம்முறையின் காரணமாக எதிர் காலத்தில் கால் நடைப் பண்ணைகளில் தொத்து நோய்கள் பரவுவதைத் தடுப்பது மட்டுமல்லாமல், அத்தகைய நோய் ஒன்று வருவதற்கு முன்னரே பல வகையான எச்சரிக்கை முறைகளையும் கைக்கொண்டு நோய் தோன்றுவதையே தடைப்படுத்தவும் இயலும்.

விஞ்ஞானம், பொறியியல்

விஞ்ஞானத் துறையில் கம்ப்யூட்டர் எவ்வாறு பயன்படுத்தப் படுகிறது என்பதை விரிவாகப் பார்க்க வேண்டுமாயின் அதற்கென்றே தனிப் புத்தகம் ஒன்று எழுதவேண்டியிருக்கும். சுருக்கமாகக் கூறப்போனால் இன்று கம்ப்யூட்டரைப் பயன்படுத்தாத விஞ்ஞானத் துறையே இல்லை என்று கூறிவிடலாம். பௌதிகம், இரசாயனம், உடற்கூறு, மருத்துவம், பொறியியல் யாவற்றிலும் இன்று ஏற்பட்டுள்ள பெரும் வளர்ச்சியும், வளர்ச்சி வேகமும் நேரிடையாகவோ அல்லது மறைமுகமாகவோ, கம்ப்யூட்டருக்குக் கடன்பட்டுள்ளன என்று கூற வேண்டும்.

வணிகத்துறை கணிப்புகளுக்கும், விஞ்ஞானக் கணிப்புகளுக்கும் அடிப்படையில் சில வேறுபாடுகள் உண்டு. வணிகத் துறைக் கணிப்புகளில் ' கணிப்பு ' (Computation) எளிதானதாக இருக்கும். கூட்டல் கழித்தல்களும், பெரும்பாலும் செய்யப்படுவதோடு கணிப்பு தீர்ந்து விடும். ஆனால் விஞ்ஞானத் துறைக் கணிப்புகள் மிகவும் சிக்கல் வாய்ந்தவை. விஞ்ஞான கணிப்புகளும் முடிவாகக் கூட்டல், கழித்தல் போன்ற அடிப்படைக் கணிப்புகளாகப் பிரிக்கப்பட்டுதான் கம்ப்யூட்டரால் செயல்படுகின்றன; என்றாலும், இத்துறையில் செய்யப்படும் கணிப்பின் அளவிற்கும், வணிகத்துறைக் கணிப்பின் அளவிற்கும் பெரிய வேறுபாடு உண்டு. வணிகத்துறையில் நூற்றுக்கணக்

கான கூட்டல், கழித்தல்களைச் செய்வதாக வைத்துக் கொண்டால், விஞ்ஞானத் துறையில் பல்லாயிரக் கணக்கான கணிப்புகளைப் பல சிக்கல்கள் நிறைந்த சூத்திரங்களைக்கொண்டு செய்ய வேண்டி யிருக்கும். ஆகையால் விஞ்ஞானத்துறை கணிப்பிற்கான கம்ப்யூட்டர் கணிப்பு வேகம் மிகுந்ததாகவும் பெரிய நினைவகத்தைக் கொண்டதாகவும் இருத்தல் அவசியமாகிறது. மற்ரெரு வேறுபாடு கொள் / கொடு விவரங்களின் அளவு ஆகும். விஞ்ஞானத் துறையில் கொள் / கொடு விவரங்களின் அளவு குறைவாகவே இருக்கும். ஆனல் வணிகத் துறையில் இதன் அளவு மிகவும் பெரியதாக இருக்கும். எடுத்துக் காட்டாக ஒரு நிறுவனத்தில் பத்தாயிரம் ஆட்கள் வேலை செய்வதாக வைத்துக் கொள்ளுவோம். இவர்களுக்கான சம்பளப் பட்டியலை ஓர் கம்ப்யூட்டர் தயாரிக்கிறது என்று வைத்துக் கொள்ளுவோம். ஒவ்வொரு தொழிலாளியின் சம்பளத்தையும் கணிப்பதற்கான 'கணிப்பு' சில கூட்டல்களும் சில கழித்தல்களும் உடைய மிக எளிதான முறையாம். ஆனல் ஆளுக்கு ஓர் கார்டு வீதம் 10 ஆயிரம் கார்டுகளை இக்கணிப்பில் துளையிட்டாக வேண்டும். இக்காரணத்தால் வணிகத்துறைக் கணிப்புகளைச் செய்யும் பொறிகளில் நினைவகம் சிறியதாயிருந்தாலும் கொள்/கொடு அங்கங்கள் (In put/out put devices) வேகமானவையாய் இருத்தல் அவசியமாகிறது.

மூன்ருவதாக உள்ள வேறுபாடு புரோகிராம்களைப் பயன்படுத்துவது பற்றியாகும். வணிகத் துறையில் ஓர் முறை ஓர் புரோகிராமைத் தயாரித்துவிட்டால் பெரும்பாலும் அதே புரோகிராமைத்தான் திரும்பத் திரும்பப் பயன் படுத்துகிருர்கள். (எடுத்துக்காட்டு: சம்பளப் பட்டியல் தயாரிக்கும் புரோகிராம் ஓர் முறை எழுதப்பட்டால் அதையே ஒவ்வொரு மாதமும் பயன்படுத்தலாம்.) ஆனல் விஞ்ஞானத் துறையிலோ பெரும்பாலும் ஓர் புரோகிராம் தயாரிக்கப்பட்டுப் பயன்படுத்தப்பட்டு முடிவடைந்தால் அதை மீண்டும் பயன்படுத்த வேண்டிய அவசியம் ஏற்படாது.

கம்ப்யூட்டர்கள் வெறும் கருவிகளாக மட்டும் பயன்படுகின்றன என்று கூறமுடியாது. கம்ப்யூட்டரின் பெருவேகம், பல்வேறு துறைகளிலும் வெறும் கணிப்புக்காக மட்டும் பயன்படுத்துவதில்லை; வணிகம், தொழில் துறைகளின் செயல் முறையிலேயே (Functioning) கம்ப்யூட்டர் பெரும் மாறுதல்களை ஏற்படுத்தியுள்ளது. அதுபோலவே விஞ்ஞானம், பொறியியல் துறைகளில் ஆராய்ச்சி முறைகளேயே கூட கம்ப்யூட்டர் மாற்றி

அமைத்துள்ளது. எடுத்துக்காட்டாகக் கணிதத் துறையில் சில கணிப்பு முறைகள் கம்ப்யூட்டர் வருவதற்கு முன்னர், கொள்கை அளவில் சிறந்த முறைகளாகக் கருதப்பட்டிருப்பினும் அவற்றை யாரும் பயன்படுத்தியதில்லை. அதன் காரணம் இம்முறைகளில் உள்ள கணிப்பின் அளவு அளவிடற்கரியதாயிருந்ததே. கணிப்புகளைக் கம்ப்யூட்டர்கள் செய்ய ஆரம்பித்த பிறகு, அதிலும் மிகை வேகக் கம்ப்யூட்டர்கள் வந்த பிறகு, ஒருகாலத்தில் கைவிடப் பட்ட முறைகள் இன்று மிக நல்ல கணிப்பு முறைகளாகப் பயன்படுத்தப்படுகின்றன.

தொழில்துறைகள்

பெரிய பெரிய தொழிற்சாலைகளில் உற்பத்தி முறைக் கண்காணிப்புப் (Process Control) பணியை இன்று கம்ப்யூட்டர்கள் செய்கின்றன. பெரிய பெரிய எண்ணெய் சுத்திகரிப்பு ஆலைகள், எஃகு தொழிற்சாலைகள், மின்சார உற்பத்தி நிலையங்கள், சிமெண்ட் ஆலைகள் போன்ற பலதிறப்பட்ட, சிக்கல் வாய்ந்த உற்பத்தி முறை கொண்ட தொழிற்கூடங்களின் உற்பத்தி முறைகளைக் கம்ப்யூட்டர்கள் கண்காணிக்கின்றன. பெட்ரோலியம் சம்பந்தப்பட்ட தொழில்துறைகளில் கம்ப்யூட்டர் பெரும் அளவில் பயன்படுத்தப்படுகின்றன. ஒவ்வொரு வாடிக்கைக்காரரும், ஒவ்வொரு மாதமும் எவ்வளவு பெட்ரோல் வாங்குகிறார் என்ற எளிய விவரத்திலிருந்து பெரிய பெரிய சுத்திகரிப்பு ஆலைகளுக்கான எண்ணெய்க் கலவை விகிதங்களைக் கணிப்பது, புதிய எண்ணெய்க் குளங்களைக் கண்டுபிடித்து எந்தெந்த இடங்களில் எத்தனைக் கிணறுகளை வெட்டி, ஒவ்வொரு கிணற்றிலிருந்தும் ஆண்டு ஒன்றுக்கு எவ்வளவு எண்ணெய் எடுத்தால் மிக அதிகப் படியான நிகர லாபம் கிடைக்கக்கூடும் என்ற விவரங்களைக் கணித்துக் கொடுப்பது ஆகியவைவரை கம்ப்யூட்டர்கள் செய்கின்றன.

தொழில்துறைகளில் வடிவமைப்பு (Design) ஓர் முக்கியப் பணியாகும். சில ஆண்டுகளுக்கு முன்வரை கம்ப்யூட்டர்கள் சாதாரணமாக வடிவமைப்பு சம்பந்தப்பட்ட கணிப்புகளை மட்டுமே செய்துவந்தன. ஓர் கட்டடத்தையோ, அல்லது இயந்திரத்தையோ வடிவமைத்த பிறகு மாற்றவேண்டுமாயின், பெரும்பாலும் மீண்டும் எல்லா கணிப்புகளையும் திரும்பவும் செய்ய வேண்டியிருக்கும்; அதுவுமல்லாமல் வடிவமைப்பில் உள்ள அடிப்படை அம்சமான வரைபடத்தை (Drawing) நேரிடையாக அன்றைய கம்ப்யூட்டர்கள் கையாள இயலவில்லை. வரைபடத்தில் உள்ள பரப்பளவு, நீள அகலங்கள் யாவும் முழு விவரங்

களுடன் எண்கள் வடிவில் புரோகிராமில் கொடுக்க வேண்டியிருந்தது. சில ஆண்டுகளுக்கு முன் கண்டுபிடிக்கப்பட்ட 'ஸ்கெட்ச் பேட்' (SKETCH PAD) போன்ற சாதனங்களால் இன்று கம்ப்யூட்டருக்கு நாம் வடிவமைப்பு சம்பந்தப்பட்ட **வரைபடங்களையே** நேரிடையாகக் கொடுக்கலாம். வரைபடங்களின் அடிப்படை அமைப்பு, நேர்க்கோடுகள், வட்டப்பகுதிகள், பரவளைவுகள் (Straight Lines, Circular Arcs, Parabolae) போன்றவையே. 'ஸ்கெட்ச் பேடில்' டெலிவிஷன் திரையைப் போன்ற ஓர் திரை உள்ளது. அந்தத் திரையின் மீது நாம் விரும்பும் கோட்டை ஓர் ஒளிப் பேனுவால் (Light pen) வரைந்தால், நமது வரைவில் உள்ள தவற்றைச் சரிப்படுத்திச் சரியான கோடாகத் திருத்தித் திருத்தப்பட்ட கோட்டை எடுத்துக்கொண்டு, கணிப்பைச் செய்கிறது. எடுத்துக்காட்டாக, P, Q என்ற இரண்டு புள்ளிகளுக்கிடையே நாம் ஓர் நேர்க்கோட்டை வரைய எண்ணுகிறோம். முதலில் 'SKETCH PAD'-ல் 'நேர்க்கோடு' என்று குறிக்கப்பட்டிருக்கும் பித்தானை அழுத்திவைத்துக்கொள்ளவேண்டும். பின்னர் ஒளிப்பேனுவின் உதவியால் P, Q வைச் சேர்த்து ஒரு கோடு வரைய வேண்டும். நாம் வரைந்த இந்தக் கோடு எவ்வளவு வளைவு உள்ளதாயிருப்பினும், ஸ்கெட்ச் பேட் இதை P ஐயும் Q வையும் சேர்க்கும் ஓர் நேர்க்கோடாகத் திருத்தி எடுத்துக் கொள்ளுகிறது. இவ்வாறே நாம் கைபோன போக்கில் ஒரு வட்டம் வரைந்தால்: ஸ்கெட்ச் பேட் அதைத் திருத்தி ஒரு முழுமை பெற்ற வட்டமாக மாற்றிக் கொடுக்கிறது. இவ்வாறு திருத்தப்பட்ட வடிவங்களின் பரிமாணங்களை உடனே, தானுகவே, கணிப்பிற்காக எண் வடிவில் கம்ப்யூட்டருக்கு அது அளிக்கிறது. இத்தகைய தன்மையால், கம்ப்யூட்டரோடு இணைக்கப்பட்ட ஸ்கெட்ச் பேடின் உதவிகொண்டு வடிவமைப்புச் செய்வதும், உடனுக்குடன் மாற்றங்கள் செய்வதும் எளிதாகிறது. இம்முறையால் வடிவமைப்பிற்காக ஆகும் காலமும் வியக்கத்தக்க விதத்தில் குறைகிறது. ஒரு குறிப்பிட்ட தொழிற்கூடத்தில் ஸ்கெட்ச் பேடின் உதவிகொண்டு வடிவமைப்பு செய்ததில், வடிவமைப்புக் காலம் 98% குறைந்தது! இதிலிருந்து இப்பொறியின் பயன் புலனுகும். 'ஸ்கெட்ச் பேட்' உதவியுடன் வடிவமைப்புச் செய்யும் தொழில்களுள் மோட்டார்த் தொழிற்சாலைகள் முதன்மையானவை.

மோட்டார்த் தொழிற்சாலைகளில் வடிவமைப்புச் செய்வது மட்டுமன்று, உற்பத்தியின் பலதிறப்பட்ட அலுவல்களுக்கும் கம்ப்யூட்டர் பொறுப்பேற்கிறது. அத்துடன் ஒவ்வொரு வகையான ஆர்டர்களுக்கும் உரிய, நிறம், கட்டுமானம், இயந்திரத்தின்

சக்தி இன்னும் பலதிறப்பட்ட விவரங்களுக்கும் ஈடு கொடுத்து வாடிக்கையாளரின் திருப்திக்கு ஏற்ப கார்களை உற்பத்தி செய்வதிலும் கம்ப்யூட்டர் பணி இன்றியமையாததாகிறது. அதே சமயத்தில் காரின் தரம் குறையாமலிருக்க கார் உற்பத்தியின் பல வேறு நிலைகளிலும், ஏற்படக்கூடிய தவறுகளைத் தவிர்க்கவும் உதவுகிறது. அமெரிக்காவில், 'டெட்ராய்ட்' நகரில் 'பிளைமவுத்' தொழிற்சாலையில் இயங்கும் கம்ப்யூட்டர் ஒரு காரின் உற்பத்தியின்போது ஏற்படக்கூடிய 3300 வகையான தவறுகளைத் தனது நினைவகத்தில் வைத்திருக்கிறது. காரின் உற்பத்தி வழியிலிருந்து 150 கண்காணிப்பாளர்கள் அனுப்பும் விவரங்களை இடைவிடாமல் இக் கம்ப்யூட்டர் ஆய்ந்து, தனக்குத் தெரிந்த 3300 வகைத் தவறுகளில் ஏதேனும் ஒன்று நடந்துகொண்டிருக்கிறதா என்று பரிசோதிக்கிறது. தவறு ஏற்படின் உடனே உற்பத்தி வழியில் (Production Line) எவ்விடத்தில் என்ன தவறு நடக்கிறது என்பதையும் உடனடியாகக் கூறுகிறது. இதனால் உற்பத்தியாகும் கார்களின் தரம் உயர்வாக இருப்பது மட்டுமன்றி, பழுதான கார்கள் உற்பத்தி செய்வது முற்றிலும் தவிர்க்கப்பட்டு, தொழிற்சாலைக்கு ஏற்படக்கூடிய நட்டங்கள் தவிர்க்கப்படுகின்றன.

அமெரிக்காவில், வங்கிகளின் வரவு செலவுத் துறைகளில் மிகப் பெரிய அளவில் இன்று கம்ப்யூட்டர்கள் பயன்படுத்தப் படுகின்றன. கம்ப்யூட்டர்கள் இல்லையாயின் அமெரிக்காவின் வங்கிகளின் இயக்கமே இன்று நிலைகுலைந்து போகும். அங்குள்ள எல்லா வங்கிகளிலும் கம்ப்யூட்டர்களை எடுத்துவிட்டால், பணம் கொடுத்து / வாங்கும் பணிக்கு மட்டுமே, அந்நாட்டில் உள்ள பெண்கள் எல்லோரும் தேவைப்படுவர் எனக் கணக்கிட்டிருக்கிறார்கள். கிட்டத்தட்ட 10 ஆண்டுகளுக்கு முன்னரே அந்நாட்டிலுள்ள 15000 பாங்குகளில் ஆண்டு ஒன்றுக்கு 20 மில்லியன் செக்குகள் புழங்கப்பட்டன; ஒவ்வொரு செக்கிற்கும் அதைக் கணக்கில் கொண்டுவந்து வரவு செலவை முடிக்க 20 வெவ்வேறு செயல்கள் தேவைப்படுகின்றன. இவைமட்டுமல்லாமல் ஓர் ஆண்டினூடே தயாரிக்கப்படவேண்டிய பல்லாயிரக் கணக்கான அறிக்கைகளையும் சேர்த்துக்கொண்டால் மொத்த வேலையின் பரிமாணம் புலனாகும்.

1950ஆம் ஆண்டு, உலகிலேயே மிகப் பெரிய தனியார் வங்கியான பேங்க் ஆப் அமெரிக்காவின் பிரசிடெண்ட் 'கிளார்க்பெய்ஸ்' என்பவர் ஸ்டான்போர்ட் ஆராய்ச்சிக் கழகத்தை அணுகி, தமக்கு ஏற்பட்டஒரு சிக்கலை விளக்கினார். அவரது வங்கி அதுவரை பயன்படுத்திவந்த கணிப்புப் பொறிகளின் வேகம் போதுமானதா

படம் 6.1

MICR—செக்

யில்லை. அப் பொறிகளுக்குப் பதிலாக, வங்கியின் பலவேறு அலுவல்களையும் கையாளவல்ல (வரவு செலவுகளைப் பார்த்துக் கொண்டு, வாடிக்கையாளரின் கணக்குகளை முறைப்படி வைத்து, அப்போதைக்கப்போது தேவைப்பட்ட அறிக்கைகளையும் தயாரித்துக் கொடுத்தும், அதே சமயத்தில் அதுவரை வாடிக்கையாளர்களின் செக்-கையாளும் முறையில் மாறுதல் ஏதும் தேவைப்படாத) ஓர் எலக்ட்ரானிக் பொறி வேண்டுமென்றனர். இத்தகைய எலக்ட்ரானிக் சாதனம் ஒன்றைத் தயாரிக்க முனைந்த ஸ்டான் போர்ட் ஆராய்ச்சிக் கழகத்தினருக்கு மிகவும் கடினமாகத் தோன்றிய பகுதி, செக்குகளைப் படிக்கவல்ல சாதனம் ஒன்றைத் தயாரிப்பதே. எத்துணையோ முறைகளை ஆய்ந்து பயன்தராது என்று கைவிட்ட பிறகு, ஒரு வழியைக் கண்டுபிடித்தார்கள். அதற்கு M I C R முறை Magnetic Ink Character Recognition) என்று பெயர். இம் முறையில் எழுதப்படும், விநோதமான வடிவம் கொண்ட எண்களையும், உருக்களையும் செக்குகள்மீது அச்சிட்டார்கள். இவ்வெண்களை அச்சிடப் பயன்படுத்தும் மையில் இரும்பு ஆக்ஸைட் இருப்பதால் இந்த எழுத்துகளை எளிதில் காந்தப்படுத்தி எலக்ட்ரான் கருவிகளைக்கொண்டு படிக்க இயலும்.

மேற்கண்ட 'MICR' முறையைப் பலவகையிலும் திருத்தி, வேறு சில பாங்குகளும் பயன்படுத்தின. இத்தகைய கருவிகளை உற்பத்தி செய்யும் எலக்ட்ரானிக் நிறுவனங்கள் செக்கைக் கையாளும் முறையை மேலும் மேலும் எளிதாக்க முயன்றுவருகின்றனர். ஆனால் காலப்போக்கில் இத்தகைய கருவிகளுக்குந் தேவையில்லாமலும் போகலாம். ஏனெனில் எதிர்காலத்தில் செக்கையோ, பணத்தையோ கையாளவேண்டிய தேவையில்லாமல் போகலாம். எதிர்காலத்தில் சம்பாதிப்பவர் ஒவ்வொருவரும் ஒரு வங்கியில் கணக்கு வைத்திருப்பவர்கள் ஒவ்வொரு வங்கியும் தனது கணக்கு வைக்கும் பொறுப்புகளைக் கம்ப்யூட்டர்களிடம் விட்டுவிடுவார்கள். ஒவ்வொரு வங்கிக்கும் உரிய சம்பளம் அவரிடம் கொடுக்கப்படாமல், அவர் வேலை செய்யும் காரியாலயத்திற்குரிய பாங்கின் கம்ப்யூட்டர், சம்பளம் வாங்குபவரின் பாங்கிக் கம்ப்யூட்டருக்கு அவரது கணக்கில் அவரது சம்பளத் தொகையை வரவு வைத்துக்கொள்ளும்படி ஆணையிடும். அதே சமயத்தில், சம்பாதிப்பவர் தாம் செலவிடும்போது காசு ஏதும் கையாளமாட்டார். அவர் வாங்கும் பொருள்களுக்கான பில்களைத் தமது பாங்கியில் கொடுத்துவிடுவார். அவரது பாங்கிக் கம்ப்யூட்டர் பில் தொகைகளை உரிய கம்பெனிகளில், (கடைகள் அல்லது நபர்களின்) கணக்கில் வரவு வைத்துக்கொண்டு, இவரது கணக்

கிலிருந்து கழித்துக்கொள்ளும். ஆ க, மனிதர்கள் நாணயங்களையோ, செக்குகளையோ கையாளும் தேவை ஏற்படாது. கொள்கை அளவில் மிக நல்ல ஏற்பாடாகவும், திறமையாகவும் உள்ள இந்த அமைப்பு எவ்வளவு காலத்திற்குப் பின்னர் ஏற்படப்போகிறது என்பதை யாரும் அறுதியிட்டுக் கூறமுடியாது. அப்படியே வந்தாலும், எல்லோரும் இதில் திருப்தியடைவார்களா என்றும் கூற இயலாது. நாணயங்களையும், நோட்டுக்களையும் கையாண்டு அதில் ஒரு தனி இன்பம் கண்ட நமது கரங்கள் இந்த முறையை எவ்வளவுதூரம் வரவேற்கும் என்று கூறமுடியாது.

விமானமும் விமானப் போக்குவரத்தும்

இன்று உலகிலுள்ள பெரிய தொழில்களுள், விமானங்களைத் தயாரிப்பதும், விமானப் போக்குவரத்துகளை நடத்துவதும் ஒன்றுகும். இத் தொழிலில் பெரும் பணம் புழங்குகிறது. விமானத்தில் உள்ள ஒவ்வோர் அம்சமும், பயணச்சீட்டின் விலை உள்பட, மிகவும் விலை உயர்ந்த ஒன்ருகும். ஆகையால் இந்தத் தொழிலின் ஒவ்வொரு துறையிலும் செலவை முடிந்த அளவு குறைப்பது இன்றியமையாதது ஆகும். இந்தத் துறையில் பெருத்த லாபத்திற்கும், பெருத்த நஷ்டத்திற்கும் உள்ள இடையீடு மிகச் சிறியது. ஒரு விமானப் போக்குவரத்துக் கம்பெனியில் நாள் ஒன்றுக்குச் சராசரி 80% இருக்கைகள் (Seats) விற்பனையாகி, அக்கம்பெனிக்குப் பெருத்த லாபம் கிடைக்கிறதென்ருல், அதே விற்பனை 75%க்குக் குறைந்தால் பெருத்த நஷ்டம் ஏற்படக் கூடும். அதனுல் எந்த அளவுக்கு அதிகப்படியான டிக்கட்டுகள் விற்கமுடியுமோ அந்த அளவுக்கு விற்ருகவேண்டும். ஆனுல் அதே சமயத்தில் இருக்கும் இருக்கைகளின் (Seats) எண்ணிக்கைகளைவிட அதிகப்படியான இருக்கைகளையும் விற்றுவிட முடியாது. இதனுல் வாடிக்கையாளருக்கு அசௌகரியம் ஏற்பட்டுக் காலப் போக்கில் வாடிக்கையை இழக்கவும் நேரிடலாம்.

ஒரு விமானப் பயணி எதிர்பார்ப்பது விரைவு, பாதுகாப்பு, கவனிப்பு ஆகிய மூன்றுமே. இம்மூன்றையும் ஓர் விமானக் கம்பெனி தனது வாடிக்கையாளருக்கு அளிக்கவேண்டுமாயின், அக்கம்பெனிக்கு இன்று கம்ப்யூட்டர் தேவைப்படுகிறது.

இந்த உண்மையை அறிந்தே இன்று உலகின் பெரிய விமானப் போக்குவரத்துக் கம்பெனிகள் யாவும் கம்ப்யூட்டரைப் பயன்படுத்துகின்றன. கம்ப்யூட்டரை மிகப் பெருமளவில் பயன் படுத்தும் கம்பெனி "PANAM" ஆகும். இதன் தலைமை நிலையம்

நியூயார்க் நகரில் உள்ளது. அங்குள்ள பெரிய கம்ப்யூட்டர் ஒன்று உலகின் 6 கண்டங்களில் உள்ள PANAM காரியாலயங்களுடன் இணைக்கப்பட்டுள்ளது. இந்தக் கம்ப்யூட்டர், விமானக் கம்பெனி யின் பலதிறப்பட்ட கணிப்புகளையும் மேற்கொள்ளுகிறது. பயணி களது ரிசர்வேஷன்களைச் செய்வது, உலக முழுவதும் பரவிக் கிடக்கும் PANAM சாமான் போக்குவரத்து (Cargo) களைக் கண் காணிப்பது, இவற்ருேடு சம்பந்தப்பட்ட, தலைமையகத்திற்கும் மற்றக் காரியாலயங்களுக்குமிடையேயான செய்திப் போக்கு வரத்துகளை மேற்கொள்ளுவது, விமானப் பயணிகளுக்காக அங்கங்கே PANAM உடன் தொடர்புகொண்ட ஹோட்டல்களில், அறைகளை ரிசர்வ் செய்வது, இவற்றுடன் வழக்கமாகச் செய்ய வேண்டிய, சம்பளப் பட்டியல் தயாரிப்பது, கணக்கு வைப்பது, கையிருப்புச் சாமான்களைக் கண்காணித்துக் கையிருப்புக் குறைந் தவுடன் தேவைப்பட்ட பொருள்களுக்கான ஆர்டர்களைத் தயா ரிப்பது, விற்பனைகள் பற்றிய புள்ளி விவரம் தயாரிப்பது ஆகிய யாவற்றையும் இந்தக் கம்ப்யூட்டர் கவனித்துக் கொள்ளுகிறது.

எடுத்துக்காட்டாக, பயணிகளுக்கு டிக்கட் விற்கும் பணியில், PANAM கம்ப்யூட்டர் எவ்வாறு வேலைசெய்கிறது என்பதைச் சற்றுப் பார்ப்போம். உதாரணமாக, பாரிஸ் நகரின் PANAM காரி யாலயத்திற்கு ஒருவர் சென்று நியூயார்க்கிற்கு ஒரு குறிப்பிட்ட தேதிப் பயணத்திற்காக டிக்கட் வேண்டுமென்கிருர். உடனே அங் குள்ள கிளார்க் ஒருவர், தன் அருகில் உள்ள ஒரு பொறியில் (Air Information Device) ஒரு பித்தானைத் தட்டி ஒரு கார்டை உள்ளே போடுகிருர். இந்தச் செயல், குறிப்பிடப்பட்டுள்ள தேதியில் பாரிஸிலிருந்து, நியூயார்க்கிற்குச் செல்ல ஒரு சீட் தேவைப்படு கிறது என்பதை ஆயிரக்கணக்கான மைல்களுக்கு அப்பால் நியூயார்க் நகரில் PANAMAC தலைமை நிலையத்தில் உள்ள கம்ப்யூட்டருக்குத் தெரிவிக்கிறது. உடனே அக்குறித்த தேதியில் எந்தெந்தப் பறப்பில் (Flight) இடமுள்ளது என்ற விவரத்தைத் தலைமைக் கம்ப்யூட்டர், பாரிஸ் காரியாலயத்திலுள்ள பொறியில் அச்சிடுகிறது. இவ்விவரங்களை, காரியாலய கிளார்க் வாடிக்கை யாளரிடம் கூற அவர் குறிப்பிட்ட பறப்பை (Flight) தேர்ந் தெடுக்கிருர். இப்பறப்பு எண்ணைக் கம்ப்யூட்டருக்கு அறிவித்து, உடனே 'Sell' என்று குறிக்கப்பட்டுள்ள பித்தானை கிளார்க் அழுத்துகிருர். உடனே நியூயார்க்கிலுள்ள கம்ப்யூட்டர் அந்தக் குறிப்பிட்ட விமானத்தில் விற்பனையான இருக்கைகளின் எண் ணிக்கையில் ஒன்றைக் கூட்டிக்கொண்டு, டிக்கட் விற்பனை செய் யப்பட்ட விவரத்தைப் பாரிஸில் உள்ள பொறியில் அச்சிடுகிறது. அதே சமயத்தில் பாரிஸ் காரியாலய கிளார்க், பயணியின்

பெயர், டெலிபோன் எண் முதலியவற்றுடன் சில தனிப்பட்ட கோரிக்கைகளையும் (போகுமிடத்தில் கார், ஓட்டல் அறை முதலியன ;ரிசர்வ் செய்வது, உணவுத் தேவைகள் போன்றவை) குறித்துக்கொண்டு அவற்றை ரிசர்வேஷன் பொறியில் டைப் செய்கிறார். இவ்விவரங்கள் யாவும் நியூயார்க் கம்ப்யூட்டருக்கு நேராகப் போய்ச் சேருகிறது. இவ்விவரங்களை டைப் செய்து முடித்தவுடன் விற்பனை முடிந்தது (End Transaction) என்ற பித்தானை கிளார்க் அழுத்துகிறார். உடனே தலைமையகத்தில் உள்ள கம்ப்யூட்டர், வேண்டிய விவரங்கள் யாவும் பெறப்பட்டனவா என்று சோதிக்கிறது. ஏதாவது விவரங்களை டிக்கட் விற்பனை செய்பவர் மறந்திருந்தால் உடனே அதை அறிவிக்கிறது. இல்லையெனில் 'O. K' என்று டைப் செய்து விற்பனையை முடிவுக்குக் கொண்டுவருகிறது. இவ்வளவுக்கும் மொத்தமாக ஆகும் நேரம் மூன்று நிமிடங்களே!

விமானப் பயணிகளின் இருக்கை சம்பந்தப்பட்டவையன்றி இன்னும் பல வழிகளிலும் விமானப் போக்குவரத்துக் கம்பெனியினரால் கம்ப்யூட்டர் பயன்படுத்தப்படுகிறது. எடுத்துக்காட்டாகப் பறப்புத் திட்டங்களைத் (Flight Plan) தயாரிப்பதைக் கூறலாம். விமானத்தின் ஜெட் என்ஜின்கள் ஏராளமாகப் பெட்ரோலைச் சாப்பிடக்கூடியவை. இரண்டு இடங்களுக்கு இடையே விமானம் பறக்கும் தூரத்தைச் சற்றுக் குறைத்தாலே பெரும் லாபம் ஏற்படக் கூடும். ஒரு குறிப்பிட்ட விமானக் கம்பெனியில் பறப்புத் திட்டங்களை ஒரு கம்ப்யூட்டரைக் கொண்டு தயாரித்ததில், கம்ப்யூட்டருக்கு ஆகும் செலவைப் போல் 30 மடங்கு செலவு பெட்ரோல் செலவில் குறைந்தது.

வானிலை முன்னறிவிப்பு

வானிலையைப் பற்றிய முன்னறிவிப்பு இன்று மிகவும் தேவையான ஒன்றாகும். புயல், பெருமழை, வரட்சி போன்றவற்றைப் பற்றி முன்கூட்டியே அறிந்துகொள்ளுவது, விவசாயத்துறைக்கு மட்டுமல்லாமல், விமானம், கப்பல் போன்ற சாதனங்களின் போக்கு வரத்திலும், போர்க்காலங்களில் பாதுகாப்புத் துறையிலும் இன்றியமையாத ஒன்றாகியுள்ளது. வானிலையை முன் கூட்டிக் கணிப்பதில், வானிலையுடன் தொடர்புகொண்ட எல்லா அம்சங்களையும் கணக்கில் எடுத்துக் கொண்டு, கணிப்பதானால் அதற்கு 64,000 ஆட்கள் தேவைப்படும். இத்தகைய ஒரு கணிப்பை ஆட்களை வைத்துக் கணித்தாலும், அவர்கள் கணித்து முடிப்பதற்குள் அவர்கள் முன்கூட்டியே அறிவிக்கவேண்டிய

வானிலே வந்து, போய்விடும். அதனல் 1956ஆம் ஆண்டுக்கு முன்வரை வானிலேயுடன் தொடர்புடைய நூற்றுக் கணக்கான அம்சங்களின் (Variables) சிலவற்றை மட்டுமே கருத்தில் கொண்டு வானிலே முன்னறிவிப்பைக் கணித்துக் கொண்டிருந் தார்கள். முக்கியமான பல அம்சங்கள் புறக்கணிக்கப்பட்டதால், இம்முன்னறிவிப்புகள் அடிக்கடி தவருகப் போனது இயல்பே. 1956ஆம் ஆண்டுமுதல் கம்ப்யூட்டர்களைக் கொண்டு வானிலே அறிவிப்புகள் தயாரிக்கப்பட்டன. இத்தகைய அறிவிப்புகள், முன் அறிவிப்புகளைக் காட்டிலும் சரியானவையாயிருந்தன.

ஒரு வானிலே நிலேயத்தின் பார்வையிலுள்ள நிலப்பரப்பின் பல்வேறு பகுதிகளிலிருந்தும், வானிலேயின் பல்வேறு நிலேகள், —வெப்பம், மழை, காற்று-வேகம், திசை இன்னும் பல விவரங் களும்—ஒரு கம்ப்யூட்டர் நிலேயத்திற்கு அனுப்பப்படுகின்றன. இக்கம்ப்யூட்டரின் நினைவகத்தில் சென்றகால வானிலே பற்றிய முழு விவரங்களும் அடங்கியிருக்கும். அவற்றுடன் இப்போது கிடைத்துள்ள விவரங்களை ஒப்புநோக்கியும், கணித்தும், வரும் நாட்களுக்கான வானிலேயைப் பற்றிய முன்னறிவிப்பைத் தயார் செய்கிறது. இவ்வாறு தயாரிக்கப்பட்ட அறிக்கைகள், 'அவ் வானிலே நிலேயத்தின் பார்வையில் உள்ள எல்லாப் பகுதிகளுக்கும் அனுப்பப்படுகின்றன. மிக விரைவான கம்ப்யூட்டர்கள் இருப்ப தால் வானிலே பற்றிய மற்ருேர் ஆராய்ச்சியும் நடைபெறுகிறது. இப்போதைய வானிலே ஆராய்ச்சிகளில் சாதாரணமாக, சென்ற கால நிலவரங்களைப் பற்றிய புள்ளி விவரங்களிலிருந்தும் தற்சமய, வெப்ப தட்ப நிலேகளிலிருந்தும், நாளை அல்லது அதற்கும் மறு நாளைய வானிலேயைக் கணிக்கிருர்கள். இதற்கு மாருக வேறு ஒரு முறையை வான் நாய்மன் (Von Naumann) கூறுகிருர். வானிலேயுடன் தொடர்பு கொண்ட பல்வேறு இயற்கை நிகழ்வு களையும், அவற்றின் உள் தொடர்புகளையும் (Internal Relation-ship) கணித சூத்திரங்களாக அமைத்து அவற்றை தீர்க்க வேண்டும் (Solve) என்பதே அம்முறையாகும். இம்முறைப்படி, முன்னறிவிப்புச் சமன்பாடுகளில் (Forecasting Equations) இப் போதைய விவரங்களை நாம் பயன்படுத்தினல் (Substitute) நாளைய நிலவரங்கள் நமக்குக் கிடைக்கும். இந்தக் கணிதச் சூத் திர முறை வெற்றி கண்டால் அது ஒரு மிகப் பெரிய கண்டுபிடிப் பாகும். ஏனெனில், இத்தகைய சூத்திரங்கள் மூலம், வானிலேயில் ஒரு குறிப்பிட்ட அம்சத்தை (Variable) மாற்றினல் அதன் உடன் விளைவுகள், பின் விளைவுகள் யாவும் உடனே நமக்குத் தெரிய வரும். சுருங்கக் கூறின் மனிதர்கள் வானிலேயைக் கட்டுப்படுத்தி ஆள அது ஒரு மிகப் பெரிய படியாக அமையும்.

மருத்துவம்

கம்ப்யூட்டர்களின் பணி மருத்துவத் துறையில் விலை மதிப் பற்றது. ஆயிரக்கணக்கான நோயாளிகளின் நோய் வரலாற்றை ஆய்ந்து மருந்துகளின் உடன் விளைவுகளை (Side Effects) கண்டு பிடிப்பதிலும், நோயைக் கணிப்பதிலும் (Diagnosis) இவற்றின் பயன் பெரிது. அதுவுமல்லாமல், அறுவைத் துறையில் (Surgery) நோயாளிகளைப் பரிசோதிப்பதும், மருத்துவக் குறிப்புகளைக் கையாள்வதிலும்கூட கம்ப்யூட்டர்கள் பயன்படுகின்றன.

எல்லாவற்றைக் காட்டிலும் நோய்காணும் பணியிலேதான் கம்ப்யூட்டரின் பயன் வியக்கத்தக்கது. நமது மருத்துவர் எவ்வாறு நமது நோய் இன்னதென்று கண்டறிந்து மருந்து கொடுக்கிருர் ? மருத்துவரது படிப்பின் அடிப்படையுடன் அவரது நீண்ட நாளைய அனுபவத்யைும் கொண்டு அவர் பல்வேறு நோய்களையும், அவற்றின் அடையாளங்களையும் (நோய்க்குறிகள் —Symptoms) மனத்தில் கொண்டிருக்கிருர். ஆனுல் எல்லாவித மான நோய்களையோ, அல்லது ஒவ்வொரு நோயின் எல்லாவித மான நோய்க்குறிகளின் (Symptoms) திண்மைகளையோ (intensities) ஒரு மருத்துவர் முற்றும் நினைவிலிருத்திக் கொள்வது என்பது இயலாத காரியம். ஆனுல் கம்ப்யூட்டரின் நினைவில் பல நூற்றுக் கணக்கான நோய்வகைகளையும், அவற்றிற்கான நோய்க் குறி களின் அடையாளங்களையும் இருத்தி வைக்க முடியும். ஒரு மருத் துவர் ஒரு நோயாளியைப் பரிசோதித்து அவரது உடல் நிலை யையும், நோய்க் குறிகளையும் கம்ப்யூட்டருக்கு அளிப்பாரே யாஞல், உடனே தனது பரந்த நினைவகத்திலிருந்தும் ஆயிரக் கணக்கான நோய் வரலாற்று விவரங்களுடன் கொடுக்கப்பட்ட குறிகளை ஒப்பிட்டு, நோயாளியின் நோய் இன்னது என்று கம்ப்யூட்டர் எளிதில் குறிப்பிடும். இந்த வகையில் பார்த்தால் மருத்துவரின் இன்றியமையாத உதவியாளனுகக் கம்ப்யூட்டர் ஆகிவிடுகிறது. ஒரு காலத்தில் மனிதர்களின் உடல்களை முற்றுமே கம்ப்யூட்டரே சோதனை செய்து நோயைக் கண்டு பிடித்து, நோய்க்கான மருந்தையும் எழுதிக் கொடுக்கும் நிலை வரக்கூடும். அந்த நிலை வரும்போது மனிதரின் நாடி, இதயத் துடிப்புகள் இரத்த அழுத்தம் இன்னும் இவைபோன்ற விவரங் களைத் தனித்தனி எலக்ட்ரானிக் சாதனங்கள் கணித்து ஒரு கம்ப் யூட்டருக்கு அனுப்பும். இவ்விவரங்களுடனும் நோயாளியின் மற்ற நோய்க்குறிகளையும் அளந்து இந்தக் கம்ப்யூட்டர் நோயைக் கண்டு கொள்ளும். நோய்க்கும், நோய்க்குறிக்கும் உள்ள தொடர்பு போன்றதே, நோய்க்கும், நோய்தீர்க்கும் மருந்திற்

கும் உள்ள தொடர்பு. ஆகையால் அதே கம்ப்யூட்டர் தனது நினைவகத்தில் ஏற்கெனவே ஏற்பட்டிருக்கும் சிகிச்சை முறையை யும் மருந்துகளையும் கம்ப்யூட்டருடன் இணைந்திருக்கும் டைப் ரைட்டரில் அச்சடிக்கும். இதே கம்ப்யூட்டர் ஒரு குறிப்பிட்ட சிகிச்சை தொடர்ந்து நடைபெற்றும் பயனில்லாமல் இருப்பின் மாற்றுச் சிகிச்சைகளையும், மாற்று மருந்துகளையும் கூற வல்லதாயிருக்கும்.

நோயாளியின் சிகிச்சையின்போது கம்ப்யூட்டரைப் பயன் படுத்தும் முறை பற்றிப் பலவித ஆராய்ச்சிகள் இன்று நடக் கின்றன. கான்சர் நோயாளிக்கு அளிக்கப்படும் கதிர்வீச்சுச் சிகிச்சையின்போது, கதிர்வீச்சின் வீரியத்தை மிகத் துல்லியமாக ஓர் எலக்ட்ரான் பொறிதான் அளந்து கொடுக்கிறது. சில ஆபத்தான சமயங்களில் சில நோயாளிகளை மிகவும் கவனத் துடன் பார்த்துக் கொள்ள வேண்டியிருக்கும். இத்தகைய நோயாளிகளின் உடலில் மிகச்சிறு உணர்முனைகளைப் (Sensors) பொருத்தி, அவர்களது உடலியக்கத்தின் வெவ்வேறு தன்மை களையும், ஒரு கம்ப்யூட்டர் கண்காணித்துக் கொண்டிருக்கும். அத்தன்மைகளுள், ஏதேனும் சிறு மாறுதல் ஏற்படினும், உடனே அதைக் கவனித்துக் கம்ப்யூட்டர் ஓர் எச்சரிக்கை அனுப்புகிறது. நோயாளி உடனடியாகக் கவனிக்கப்படுகிறுன். இம்முறையைச் சோதிக்கச் சில ஆண்டுகளுக்கு முன்னர் இங்கிலாந்திலிருந்த ஒரு நோயாளியின் மூளையலைகள் (Brain waves) ஒரு செய்தித் தொடர்புச் செயற்கோள் மூலம் (Communication Satellite) அமெரிக்காவில் 'மினிசோட்டா'வில் உள்ள ஓர் ஆய்வு நிலை யத்திற்கு அனுப்பப்பட்டன. அவ்வலைகளை மினிசோட்டா நிலையம் பெற்று, ஆய்ந்து, திரும்பவும் இங்கிலாந்திற்கே திருப்பி யனுப்பியது. இதற்கான மொத்த நேரம் ஒரே ஒரு நிமிடமாகும்.

அவசரக் காலங்களில் கம்ப்யூட்டரின் அளவற்ற வேகம் மிகவும் பயனுள்ளதாயிருக்கிறது. சில சமயம் அசாதாரண பிரிவு களைச் சேர்ந்த இரத்தம் கிடைக்கக்கூடிய இடத்தையோ அல்லது அந்த இரத்தத்தைத் தானம் செய்யத் தயாரா யிருப்பவரின் முகவரிகளையோ பெறுவது மிகவும் அவசியமாய்ப் போய்விடும். பல்வேறு இரத்த நிலையங்களின் இரத்த இருப்புகளை வகைவாரி யாகக் கம்ப்யூட்டர் ஒன்று வைத்துக் கொண்டிருக்குமாஞல் அதன் உதவி கொண்டு எந்தப்பிரிவு இரத்தம் எந்தெந்த நிலை யங்களில் கிடைக்கும் என்பதை அறிந்து கொள்ளலாம். அதே சமயத்தில், கையிருப்பு தீர்ந்து போயிருக்குமாஞல், கம்ப்யூட் டரின் நினைவகத்திலிருக்கும் இரத்த தானம் செய்பவர்களின்

பட்டியலிலிருந்து குறிப்பிடட வகை இரத்தம் உடையவர்களைத் தேர்ந்தெடுத்து அவர்களை உடனடியாக வரவழைக்கலாம்.

மனித உடற்கூறுகள் பற்றிய ஆராய்ச்சியிலும் கம்ப்யூட்டர் பெரிதும் பயன்படுகிறது. எவ்வாறு கட்டடங்களையும், பாலங்களையும், இயந்திரங்களையும் வெறும் எண்களைக் கொண்டே கம்ப்யூட்டர் கணித்து ஆய்கிறதோ அதே அடிப்படையில் மனிதனின் உடற்கூற்றையும் அதன் இயக்கத்தையும் வெறும் எண்களாக மாற்றிக் கம்ப்யூட்டரால் ஆராய முடியும். இதையே மாற்றிக் கூறினல் கம்ப்யூட்டர், மனிதனின் உடற்கூறு, அதன் இயக்கம் இவற்றின் மாதிரிகளை வெறும் எண்களால் தயாரித்து ஆயவல்லது. இத்தகைய 'எண்—மனிதன்' மீது நாம் சிறிதும் கவலைப்படாமல் எத்தகைய பரிசோதனை வேண்டுமானுலும் ஆபத்தில்லாமல் செய்யலாம்; ஏனெனில் நம் சோதனைத் தோல்வியில் முடிந்தாலும் 'இறந்து' போகப்போவது ஒரு 'எண் மனிதனே.' எலக்ட்ரானிக்ஸ், உடற்கூறு ஆகிய இரண்டும் சேர்ந்த ஒரு புதியதுறை இப்போது உருவாகியுள்ளது. பயானிக்ஸ் (Bionics) என்ற இந்தப் புதிய துறை பொறியியலாளரையும், உடற்கூற்று வல்லுனரையும் ஒன்று சேர்க்கிறது. இந்தத் துறையில் உயிரினங்களின் இயக்கங்களை, உயிரற்ற பொறிகளில் வடிப்பதே முக்கிய நோக்கமாகும். அமெரிக்காவில் மட்டும் 100-க்கு மேற்பட்ட ஆய்வுக் கூடங்கள் இந்தத் துறையில் ஆராய்ச்சிகளை நடத்துகின்றன.

மனித மூளையின் இயக்கங்களை ஒட்டி இயங்கவல்ல கம்ப்யூட்டர்களை வடிவமைப்பது கம்ப்யூட்டர் துறையின் முன்னேடியும், பெரிய கணித மேதையுமான ஜான் வான் நாய்மனின் கனவாக இருந்தது. மனித நரம்பு மண்டலத்தையும், மூளையையும் ஆராய்வதற்கு இன்று உதவும் கம்ப்யூட்டரே, இந்த ஆராய்ச்சியினுல் பெரும் பயன் அடையக்கூடும். மூளையின் பலவேறு பாகங்களையும், அவற்றின் இயக்கங்களையும் நன்கு ஆய்ந்து தெளிந்த பின்னர், இவ்வாராய்ச்சியின் பயனுக ஒருவேளை கம்ப்யூட்டரின் அடிப்படை அமைப்பிலேயே மாறுதல்கள் ஏற்படக்கூடிய நிலை ஏற்பட்டாலும் ஏற்படலாம். மனித மூளை எவ்விதம் செய்திகளைச் சேகரித்து வைத்துக் கொள்கிறது; எவ்விதம் வேண்டும்போது உரிய செய்தியை நினைவு கூர்கிறது. அல்லது எவ்விதம் மறதி ஏற்படுகிறது என்பன போன்றவற்றுள் இன்று நமக்குத் தெளிந்த அறிவு இல்லை. இவை யாவும் ஆராய்ந்து வெற்றி காணும்போது கட்டாயம் கம்ப்யூட்டர்

அடிப்படை அமைப்பிலும் மாறுதல்களும் முன்னேற்றங்களும் ஏற்படும்.

உற்பத்தி நிறுவனங்கள், வாணிபம்

எந்த ஓர் உற்பத்தி நிறுவனத்திலும் பல வேறு அங்கங்கள் இருப்பது இயல்பே. மூலப் பொருள், கொள்முதல் செய்யும் பிரிவு (Purchase section), உற்பத்தித்திட்டப் பிரிவு (Production planning section), உற்பத்திப் பிரிவு (Production), ஸ்டாக் பிரிவு, அனுப்பும் பிரிவு (Despatch section), விற்பனைப் பிரிவு (Marketing Dept.) ஆகிய இப்பல பகுதிகளும் ஒன்றுடன் ஒன்று இணைந்து செயல்படல் மிகவும் அவசியமாகும். இத்தனைப் பிரிவுகளுக்கும் இடையே உள்ள செய்திப் போக்குவரத்தின் துரிதம், சரளம் ஆகியவற்றைப் பொறுத்தே இந்த நிறுவனம் திறமை வாய்ந்த ஒருங்காகச் செயல்பட முடியும். எடுத்துக்காட்டாக ஒரு குறிப்பிட்ட பொருளுக்கு, விற்பனைச் சந்தை குறைகிறது என்றால் அச் செய்தி விற்பனைப் பிரிவில் இருந்து உடனடியாக உற்பத்திப் பிரிவு, ஸ்டாக் பிரிவு ஆகியவற்றிற்கு அனுப்பப்பட்டு அப் பொருளின் உற்பத்தியை மட்டுப்படுத்த வேண்டும். அதே நேரத்தில் மூலப் பொருள் கொள்முதல் பகுதிக்கும் இச்செய்தி அறிவிக்கப்படவேண்டும். அப்போதுதான் இந்தக் குறிப்பிட்ட பொருளின் உற்பத்திக்குத் தேவையான மூலப் பொருளைக் கொள் முதல் செய்வதை நிறுத்தவோ, குறைக்கவோ முடியும்; இதே முறையை எதிர்வகையில் பயன்படுத்தினல், சந்தையில் ஏற்படும் புதிய அதிகப்படித் தேவைகளை (Demands) உடனே பயன் படுத்தி லாபம் சம்பாதிக்க இயலும். இந்த இரண்டிற்குமே மார்க்கெட் நிலவரத்தை உடனடியாகக் கணித்து மிகக் குறைந்த காலத்தில் தொழில் நிறுவனத்தின் மற்றப் பகுதிகளுக்கு உரிய செய்திகளை அனுப்புதல் மிக அவசியமாகும். இங்கே கம்ப்யூட்டர் இன்றியமையாததாகிறது.

இன்னேர் எடுத்துக்காட்டையும் பார்ப்போம். நாட்டில் சாமான்கள் போக்குவரத்தில் இரயில்வே பெரும் பங்கு வகிக்கிறது. ஆங்காங்கே பொருள்கள் இரயில் நிலையங்களில் சேமித்து வைக்கப்பட்டுச் சாமான் வண்டிகளில் (Goods wagons) அனுப்பப்படுகின்றன. ஓர் நிலையத்தில் எத்தனைப் பெட்டிகள் (Goods wagons) தேவைப்படும் என்பதைப்பற்றி முன் கூட்டியே ஓரளவுக்காவது அறிதல் அவசியம். தேவைக்கதிகமான பெட்டிகளை ஓர் நிலயத்தில் நிறுத்தி வைத்திருப்பின் அது பயனற்றுப் போகிறது. அதே சமயத்தில் தேவைக்குக் குறைவான பெட்டி

களை நிறுத்திவைப்பின் சாமான் போக்கு வரத்தில் தேக்க மேற்படலாம்.

எந்தெந்த நிலையங்களில் எத்தனைப் பெட்டிகளை நிறுத்தி வைக்கவேண்டும் என்ற கணிப்பு சுலபமானதல்ல. மேலும் இந்தக் கணிப்பு நிலையானதாகவும் இருக்கப் போவதில்லை. குறிப்பிட்ட காலங்களில் சில நிலையங்களில் அதிகப் பெட்டிகள் தேவைப் படலாம் (எடுத்துக் காட்டாக நெல் அறுவடைக் காலங்களில் தஞ்சை ரெயில்வே நிலையத்தில் ஏராளமான சாமான் பெட்டிகள் தேவைப்படும்). மற்றக் காலங்களில் அவ்வளவு தேவையிருக்காது. இத்தகைய வேறுபாடுகளை உடைய இக்கணிப்பைத் திறம்படச் செய்ய ஓர் கம்ப்யூட்டரால்தான் இயலும். இது வெறும் சாமான் போக்கு வரத்துக்கு மட்டுமன்று; மக்கள் போக்கு வரத்துக்கும் இதே கணிப்புப் பொருந்தும்.

விஞ்ஞானத் துறையில் கம்ப்யூட்டர் வந்த பிறகு பல புதிய கணிபுப் முறைகள் தோன்றலாயின என்று பார்த்தோம். இந்தக் கூற்றுத் தொழில் வணிகத் துறைகளுக்கும் பொருந்தும். இன்று தொழில் துறையில் பெரிதும் புழங்கும் லீனியர் புரோகிராமிங் (Linear Programming) பெர்ட் / சி. பி. எம். (PERT / CPM) ஆகிய முறைகள் கம்ப்யூட்டர்கள் இல்லையாயின் சிறிதும் பயன்படா. நாம் முன்னர்ப் பார்த்த போக்குவரத்துக் கணிப்பு களையும், சில கலவைக் கணிப்புகளையும் (சிமெண்ட் செய்வதற் கான மூலப் பொருள்களின் கலவை விகிதம் போன்றவை) எண்ணெய்ச் சுத்திகரிப்புக் கூடங்களில் கையாளப்படும் எண்ணெய் வகைகளின் கலப்பு விகிதாசாரம் ஆகியவை போன்றவைகளை லீனியர் புரோகிராமிங் கணிக்கிறது. பெரிய நிர் மாண வேலைகள், உற்பத்தி முறைகள் முதலியவற்றில் குறைந்த காலத்தில் செய்து முடிக்க வழி வகுப்பதுடன், நிர்மாணத்தின் போது ஒவ்வொரு படியிலும் நிர்மாணம் எவ்வளவு தூரம் நடந் திருக்கிறது, உரிய காலத்தில் முடிந்துவிடுமா ? அவ்வாறு இல்லையாயின் எந்தெந்த வேலைகளை எவ்வளவு துரிதப்படுத்த வேண்டும் என்பன போன்ற விவரங்களை பெர்ட் / சி. பி. எம். (PERT / CPM) கொடுக்கிறது.

கம்ப்யூட்டர் கைவசம் இருப்பதாலேயே இன்று பல வணிகத் துறைகள் வெறும் மனோயூகத்தில் ஓடாமல் விஞ்ஞான முறைப்படி நடத்தப்படுகின்றன.

இன்றைய தொழில், வாணிபத் துறைகள் யாவும், செலவைக் குறைத்து, லாபத்தைப் பெருக்கி, தரம் குறையாமல் பொருள்

உற்பத்தி செய்வதைக் குறிக்கோளாகக் கொண்டுள்ளன. இவற்றிற்கென்றே தனிப்பட்ட, மேற்கணிப்பு முறைகள் (Optimisation Techniques) உள்ளன. அம்முறைகளைப் பயன் படுத்திக் கணிப்புகளைச் செய்யவும் கம்ப்யூட்டர்கள் இன்றி யமையாதனவாய் அமைந்துள்ளன.

விண்வெளி ஆய்வு

கம்ப்யூட்டர் இல்லாமல் விண்வெளியில் செயல்படுவது இயலாது என்று நாசா (NASA-National Aeronautics and Space Administration) கூறுகிறது. விண்வெளிப் பயணத்தின் ஒவ்வொரு படியும் கம்ப்யூட்டரின் துணையுடன்தான் செயல்படுகிறது. விண்வெளிப் பயணத்தின் அடிப்படைத் தத்துவம், வடிவமைப்புகள், கட்டுமானங்கள், பரிசோதனைகள், பயணமேற்பதற்கு முன்னர் நடக்கும் சோதனைகள், விண்கோளைச் செலுத்தல், அதை விண்பாதையில் சேர்த்தல், சேர்த்தபின் பாதையில் நிலைப் படுத்திக் கட்டுப்பாட்டிற்குக் கொண்டுவருதல், வழிநடத்தல், பயணம் முடிந்த பின்னர் பூமியின் காற்று வெளியில் நுழைதல், பூமிக்கு வந்துசேர்ந்த கோளைத் தேடிக் கொணரல், கோளின் பயணத்தில் கிடைத்த விவரங்களை ஆராய்தல் ஆகிய இவ்வளவும் கம்ப்யூட்டரின் துணையைக் கொண்டுதான் நடைபெற்றாக வேண்டும்.

விண்வெளிப் பயணத்தில் இரண்டு வகையான கம்ப்யூட்டர்கள் பயன்படுத்தப்படுகின்றன. ஒன்று பூமியிலிருக்கும் கம்ப்யூட்டர்கள்; மற்றவை விண்வெளிக் கலங்களிலேயே பொருத்தப்பட்டு இயங்குபவை. இந்த இரண்டாம் வகைக் கம்ப்யூட்டர்கள் ஒரு புதிய கம்ப்யூட்டர் குடும்பத்தையே உருவாக்கிவிட்டன. விண்வெளிக் கலங்களில் எப்போதும் இடநெருக்கடி அதிகம். அநாவசியமாக இடத்தை அடைத்துக்கொள்ளும் எந்தப் பொருளுக்கும் அங்கு இடமில்லை. ஒரு பவுண்டு எடையுள்ள பொருளை விண்வெளியில் செலுத்த 1000 பவுண்டு அழுத்தம் (Thrust) தேவைப்படுகிறது. அதனால் விண்வெளிக் கம்ப்யூட்டர்கள் அளவில் சிறியதாய், எடையில் குறைந்ததாய் இருக்க வேண்டிய அவசியம் ஏற்பட்டது. அதுவுமல்லாமல் ஒரு விண் பயணத்தின் குறிக்கோள்கள் முழுவதும் இக்கம்ப்யூட்டரால் நிறைவேற்றப்படவேண்டும். ஆனால் எந்தப் பயணத்திலும் குறிக்கோள்கள் முழுதும் திட்டவட்டமாக அறுதியிட்டு திட்ட நிலையிலேயே (Planning Stage) கூறமுடியாது. கோள் செல்ல வேண்டிய விண்பாதை, கோளை வழிநடத்தும் (Guidance) முறை ஆகியவை அறுதியாகத் தீர்மானிக்கப்பட்ட பிறகு பயணத்திற்கு

வேண்டிய கம்ப்யூட்டரை வடிவமைப்பது என்றால் அநாவசிய மான காலதாமதம் ஏற்படும். இக்காரணத்தால் ஒரு பொதுப் பயன் (General Purpose) கம்ப்யூட்டர் உருவாக்கப்படுகிறது. இதன் நினைவகம், சாதாரணமாக வேண்டிய நினைவகத்தைப் போல 3–லிருந்து 5-மடங்கு பெரிதாக இருக்குமாறு அமைத்துக் கொள்ளுகிறார்கள். அதனால் ஆரம்பத் திட்டத்தில் சேர்க்கப் படாமல் விட்டுப்போன, ஏதாவது ஒரு புதிய கணிப்பைத் திடீரென சேர்த்துக்கொள்ளவும் போதிய வசதி இருக்கிறது.

இது மட்டுமல்லாமல் விண் வெளியில் செல்லும் கம்ப்யூட்டர், தரைக் கம்ப்யூட்டர்களைக் காட்டிலும் வலிமை மிக்கதாக இருத்தல் அவசியம். விண்வெளியின் கதிர்வீச்சு (Radiation) விண் கற்களின் தாக்குதல், வெற்றிடம் (Vacuum) ஈர்ப்பின்மை (O-gravity) ஆகியவற்றால் ஏற்படக்கூடிய சிக்கல்கள், பயணத் திற்கு முன்னர்ச் செய்யப்படும் தூய்மை முறைகள் (Sterilization methods), ராக்கெட்டைச் செலுத்தும்போது ஏற்படும் பெரும்அதிர் வுகள் முதலியவற்றையும், பயணம் முடிந்துத் திரும்பும்போது ஏற்படக்கூடிய ஈர்ப்பு அதிகரிப்பையும் (gravity build up) தாங்கி யாக வேண்டும். எல்லாவற்றிற்கும் மேலான சிக்கல், கம்ப்யூட்டர் கெட்டுப்போய் ஏற்படக்கூடிய அபாயம். இதைத் தவிர்க்க ஒரே சமயத்தில் 3 கம்ப்யூட்டர்களைப் பயன்படுத்துகிறார்கள். இதன் காரணமாக ஒன்று அல்லது இரண்டு கம்ப்யூட்டர்களிலும் கோளாறு ஏற்படினும், மூன்றாவது கம்ப்யூட்டராவது இயங்கும். இம்முறைப்படி 99% கம்ப்யூட்டரின் பணி தொடர்ந்து பழுதின்றிக் கிடைக்கும் என எதிர்பார்க்கலாம்.

இத்தகைய கம்ப்யூட்டர்கள் விண்வெளிப் பயணம் மட்டு மின்றி, செயற்கைக் கோள்களின் வழியாகச் செய்தித் தொடர்பு ஏற்படுத்திக் கொள்ளுவதிலும் பயனுள்ளதென நிரூ பிக்கப் பட்டுவிட்டது. டெல்ஸ்டார் (Telstar) செய்திக்கோளு டன் ஹனிவெல்-800 கம்ப்யூட்டர் ஒன்றை இணைத்து ஓர் கம்ப்யூட்டரிலிருந்து மற்றொரு கம்ப்யூட்டருக்குத் தொடர்பு ஏற்படுத்த முயற்சித்து வெற்றியும் கண்டுள்ளனர். எதிர்காலத் தில் கம்ப்யூட்டரும், விண்வெளி விஞ்ஞானமும் சேர்ந்து செய்திப் போக்குவரத்தில் மிகப் பெரும் புரட்சியை உண்டு பண்ணும் என எதிர்பார்க்கலாம்.

விண்ணாய்வுக் கோள் நிலையம் (Orbiting Astronomical Observatory) கம்ப்யூட்டரைப் பயன்படுத்தும் ஓர் ஆளற்ற விண்கலம் ஆகும். பூமிக்கு மேலே 500மைல் தூரத்தில் இக் கோள் நிலையம் பூமியைச் சுற்றி வருகிறது. களத்தில் உள்ள

கம்ப்யூட்டர் ஒன்று அதோடு இணைக்கப்பட்டுள்ள தொலை நோக்கியை விண்வெளியில் குறிப்பிட்ட திசைகளில் திருப்பி அங்குள்ள விண்மீன்களை நோக்கி இருக்குமாறு அமைக்கிறது. தொலைநோக்கி தான் காண்பனவற்றை உடனுக்குடன், 200,000 பிட் கொண்ட ஒரு ஐ. பி. எம். கம்ப்யூட்டரின் நினைவகத்தில் சேர்த்து வைத்துக் கொள்ளுகிறது. இக்கோள் பூமியைச் சுற்றிவர ஏறத்தாழ 100 நிமிடங்கள் ஆகும். இந்த நூறு நிமிடங்களில் தான் சேகரித்த எல்லா விஷயங்களையும், இக் கோள் தன்னுடன் தொடர்பு கொண்ட தரைக் கட்டுப் பாட்டு நிலையங்களின் நேர் உயரே (அமெரிக்காவில் உள்ளவை) செல்லும்போது கொட்டித் தீர்த்து விடுகிறது. இதற்கு ஆகும் நேரம் $7\frac{1}{2}$ விநாடிகளாம். இத் தரை நிலையங்கள் இச் செய்தி களைப் பெற்ற பின்னர் மேலும் அதே பரிசோதனையைச் செய்யவோ அல்லது புதிய பரிசோதனைகளை மேற்கொள்ளவோ விண் கோளில் உள்ள கம்ப்யூட்டருக்கு மீண்டும் ஆணையிடலாம்.

ஆள் கோள் விண்வெளிப் பயணங்களில் (Manned Space Flight) பயன் படுத்தப்படும் கம்ப்யூட்டர்கள் சில அடிப்படை வேறுபாடுள்ளவை. இவற்றுள் முதன்மையானது கலத்தில் உள்ளோரே இக்கம்ப்யூட்டரின் இயக்கத்தைக் காத்துக்கொள்ளும் வசதியாம் (on board maintenance). இக்கலங்களில் பயன்படுத் தப்படும் கம்ப்யூட்டருக்குத் தனித்தன்மைகள் சில உண்டு. தன்னைத் தானே சோதித்துக் கொண்டு தனது பாகம் ஏதாவது ஒன்றில் பழுது ஏற்பட்டோ அல்லது ஏற்படும் நிலையிலேயோ இருந் தால் அந்தப் பாகத்தைப்பற்றிக் கலத்திலுள்ள மனிதர்களுக்கு இந்தக் கம்ப்யூட்டர் தெரிவிக்கும். உடனே கலத்தில் உள்ளோர் குறைப்பட்ட இப்பகுதியைச் சரிசெய்வதற்குப் பதிலாக அதை முற்றிலும் நீக்கிவிட்டுப் புதிய பகுதியைச் சேர்த்துவிடுவார்கள். கம்ப்யூட்டர்களில் பழுதுபட்ட பாகத்தைச் சரி செய்வதைவிட அதை அகற்றிவிட்டுப் புதிய பாகத்தைச் செருகிவிடுவது லாபகர மானதாகும். இத்தகைய முறையும் இன்ன பிற முன்னேற்றங்களும் கம்ப்யூட்டரின் நம்பிக்கைக் காலத்தை (Reliability Period) பல மடங்கு அதிகரித்துள்ளன. எதிர்காலத்தில் மாதக் கணக்கில் மனிதர்கள் விண் வெளியில் பயணம் செய்ய வேண்டியிருக்கும். அப்போது இத்தகைய நம்பிக்கை வாய்ந்த கம்ப்யூட்டர்கள் இன்றியமையாததாகும்.

பொதுவாக ஆள்கோள் விண்-கலங்களில், பெரிய நினைவகம் கொண்டதும், நானுவிதக் கணிப்புகளையும் செய்யக்கூடியதுமான கம்ப்யூட்டர் தேவைப்படுகிறது. ஆள்கோள் பயணங்களின்

அடிப்படைத் தத்துவங்களில் ஒன்று, கலத்தினுள்ளோர் அவசிய மானல், தரைக் கட்டுப்பாட்டு நிலையத்தின் உதவியின்றித் தாமே இயங்கவேண்டும் என்பதே. கலத்தை வழிநடத்திச் செல்லுவதற் கான கணிப்புகள் கலத்திலேயே கணிக்கப்படுகின்றன. இது மட்டுமின்றி, விண் பாதையில் மாற்றங்கள் ஏற்படுத்தவும், கலத் தின் வேகத்தைக் குறைக்கவும், கூட்டவும் மற்றக் கலங்களுடன் இணைவதற்கான ஏற்பாடுகளைச் செய்தல் ஆகிய பல செயல்களும் பெருமளவுக்குக் கம்ப்யூட்டரை நம்பியே நடக்கின்றன.

ஆள் கல, விண் பயணங்களில் தரைக் கட்டுப்பாட்டு நிலையங்களில் உள்ள கம்ப்யூட்டர்கள் 'ரியல் டைம்' (Real Time) கம்ப்யூட்டராக இருக்கவேண்டும். ஏனெனில் இத்தகைய பயணத் தில் அடுத்த நிமிடம் என்ன செய்யவேண்டும் என்பதை, சென்ற நிமிட நிகழ்வுகளை ஆய்ந்து, அதன் விளைவுகளைக் கொண்டே தீர்மானிக்க வேண்டியிருக்கும். எடுத்துக் காட்டாக விண் கலத்தை, விண் பாதையில் செருக (inject) வேண்டுமா அல்லது பயணத்தை ரத்து செய்யவேண்டுமா என்பதை 25 விநாடிகளில் தீர்மானித்தாக வேண்டும். உதாரணத்திற்கு, 'மெர்க்குரி' பயணத்தை எடுத்துக்கொள்ளுவோம்.

மெர்க்குரி பயணத்தின் உயிர்நாடி வாஷிங்டன் அருகே அமைந்த கொடார்ட் விண் பயண நிலையம் (Goddard Space Flight Centre) ஆகும். கென்னடி முனையின் ஏவு தளத்திலிருந் தும் (Launch Site) உலகின் பல பாகங்களில் அமைந்திருக்கும் வான் கோள் தேடித் தொடரும் நிலையங்களிலிருந்தும் (Space Tracking Stations) வரும் ஏராளமான விவரங்கள் ஒரு I. B. M.-7094 கம்ப்யூட்டருக்கு ஊட்டப்படுகின்றன. இந்த விவரங்களை விழுங்கிச் சீரணித்துக் கம்ப்யூட்டர் உமிழும் செய்திகளைக் கொண்டு தான் பயணத்தைப் பற்றிய மிக முக்கியமான முடிவுகள் எடுக்கப் படுகின்றன.

விண்வெளிப் பயணத்தின்போது, பயணிகளின் உடல்நிலை, விண்கலத்தின் செயற்பாடுகள் ஆகியவை பற்றிய பல விவரங் களும், கலத்திலிருந்து மிக அருகில் அமைந்திருக்கும் கோள் தொடரும் நிலையத்திற்கு அனுப்பப்படுகின்றன. அங்கிருந்து இச்செய்தி உடனடியாக கொடார்டு வான்பயண நிலையத்திற்குத் திருப்பி அனுப்பப்படுகிறது (Relayed). அந்தக் குறிப்பிட்ட நொடியில் விண்கலத்தின் நிலை என்னவாயிருக்க வேண்டும் என்பதை ஏற்கெனவே கணித்துக் கம்ப்யூட்டரின் நினைவகத்தில் வைத்திருக்கிறார்கள். இந்தக் கணிப்பு முடிவோடு, இப்போது விண்கலம் இருக்கும் உண்மையான நிலையை உடனடி

யாக ஒப்பிட்டுப் பார்க்க முடிகிறது; விண்வெளிப் பாதையின் ஒவ்வொரு புள்ளியிலும் கம்ப்யூட்டரின் கணிப்பிற்கும், கோளின் உண்மையான இடத்திற்கும் அதிகப்படி வேறுபாடு 300 கஜ தூரமே. இத்தகைய துல்லிய கணிப்பின் பயனுக விண்வெளிப் பயணிகளின் கோள்கள் பூமியில் வந்திறங்கும் இடத்திற்கும், அவர்களை எதிர்பார்த்து, ஏற்கெனவே காத்துக் கொண்டிருக்கும் இடத்திற்கும் இடையே அதிகபட்சம் 4 அல்லது 5 மைல் வித்தியாசமே இருக்கிறது.

அமெரிக்காவின் நிலவுப் பயணங்களில் சம்பந்தப்பட்ட கலங்களை விண்ணில் செலுத்துவதற்குப் பயன்படுத்தும் ராக்கெட் சாட்டர்ன் V ஆகும். இது மிக வலிமை வாய்ந்த ராக்கெட் ஆகும். 7½ மில்லியன் பவுண்டு அழுத்தம் (Thrust) அளிக்கவல்ல இந்த ராக்கெட்டில் 2,00,000-க்கு மேற்பட்ட பாகங்கள் உள்ளன. பயணம் தொடர்வதற்குச் சற்று முன் இத்தனை பாகங்களும் சரியாக உள்ளனவா என்று சரிபார்த்துக் கொள்ளுதல் மிக முக்கியமாகும். இத்தனை பாகங்களும் சில விநாடிகளில் பரிசோதித்துப் பயணத்திற்கு அனுமதி கொடுக்க வேண்டுமாயின் அது கம்ப்யூட்டரால் மட்டும்தான் முடியும்.

நிலவுப் பயணங்களில் பயன்படுத்தப்பட்ட "அப்பல்லோ" கலங்களை வழிநடத்தியது ஒரு கம்ப்யூட்டரே. அப்பல்லோ கலம் பூமியின் கோளாகச் சுற்றிவரும்போது அதிலுள்ள பயணிகள், விண்மீன்களின் நிலைகளைச் சோதித்து அவ்விவரங்களைக் கம்ப்யூட்டருக்கு அளிக்கிறார்கள். உடனே கம்ப்யூட்டர் உரிய கணிப்புகளைச் செய்து எப்போது எந்த நிலையில் தற்போதைய பயணப் பாதையை விட்டு நிலவுப் பாதையில் புகுவதற்காகச் சிறிய ராக்கெட்டுகளை எரிக்க வேண்டும் என்று கூறுகிறது. நிலவை நெருங்கிய பின்னர், விண்கலம் நிலவில் இறங்கு முன்னர் நிலவுக் கோளாக மாறி நிலவைச் சுற்றிவரவேண்டியுள்ளது. அதற்காக உள்ள 'பிரேக் ராக்கெட்டுகளை' (Breaking Rockets) எந்த நேரத்தில் எவ்வளவு நேரம் எரியவிட வேண்டும் என்பதைத் தீர்மானிப்பதும் கம்ப்யூட்டரே. அதற்குப் பின்னர் நிலவைச் சுற்றிவரும் தாய்க்கலத்திலிருந்து பிரிந்து நிலவை நோக்கிச் செல்லும் சேய்க்கலத்தையும் கம்ப்யூட்டரே கண்காணிக்கிறது. நிலவுப் பரப்பு ஆய்வு முடிந்து சேய்க்கலம் நிலவிலிருந்து புறப்பட்டுத் தாய்க்கலத்தில் வந்து சேரும்வரை கம்ப்யூட்டர் அதன் ஒவ்வோர் இயக்கத்தையும் கட்டுப்படுத்துகிறது. பூமிக்குத் திரும்பி வரும்போது பூமியின் காற்றுமண்டலத்தில் நுழைய வேண்டிய இடம், கோணம் ஆகியவற்றையும் கம்ப்யூட்டர்

சரிபார்த்துக் கொடுக்கிறது. இவ்வாறாகக் கம்ப்யூட்டர் இல்லையா யின் இன்றைய விண்பயணம் இல்லை என்ற நிலைக்கு விண் ஆய்வு, விண் பயணத் துறையில் அது முக்கியத்துவம் பெற்று விட்டது.

கல்வி

கல்வித் துறையில் கம்ப்யூட்டருக்குப் பெரும் வாய்ப்பு உள்ளது. பெருமளவுக்கு இன்று கம்ப்யூட்டர்கள் இந்தத் துறை யில் பயன்படுத்தப்படாவிடினும் எதிர்காலத்தில் வியக்கத்தக்க முறையிலும், அளவிலும் கம்ப்யூட்டர் கல்வித்துறையில் பயன் படும் என்று எதிர்பார்க்கலாம்.

ஸ்கின்னர் (Skinner, Professor of Psychology at Harvard) கூறுகிறார். 'உலகின் மக்கள்தொகை இதுவரை கண்டிராத அளவுக்குப் பெருகியுள்ளது. இம்மக்கள் தொகையில் இதுவரை இல்லாத பெரும் விகிதத்தினர் கல்வி கற்க விரும்புகிறார்கள். இந்தத் தேவையை நிறையப் பள்ளிக்கூடங்களைத் திறந்து வைத்து மட்டுமோ அல்லது ஏராளமான ஆசிரியர்களைப் பயிற்று விப்பதால் மட்டுமோ பூர்த்தி செய்துவிட முடியாது. மாறாகக் கல்வி கற்பிக்கும் முறையில் திறமை அதிகரித்தாக வேண்டும்.' இதன் பொருள் என்ன?

கல்வி பெற விரும்புவோர் இன்று பலதிறப்பட்ட மக்க ளாக உள்ளார்கள். வேறுபட்ட சூழ்நிலைகளிலிருந்து வரும் சிறு வர்களின் கற்கும் திறனும், இயல்பும் மிகவும் மாறுபட்டவை யாயுள்ளன. நாற்பது ஐம்பது மாணவர்களை, ஒன்றாகக் கூட்டி வைத்துக்கொண்டு ஒருமணி நேரம் ஓர் தலைப்பைப் பற்றி ஆசிரியர் பேசித் தீர்த்துவிடுவதாலோ அல்லது கரும் பலகையில் எழுதிப்போட்டுவிடுவதாலோ, அந்தத் தலைப்பைக் கற்பித்துவிட்டதாகக் கூறமுடியாது. திறமையுள்ள மாணவனுக்கு எப்படிக் கற்பித்தாலும் அவன் கற்றுக்கொள்ளப்போகிறான். ஆசிரியரின் கவலை திறமைசாலி மாணவர்களைப் பற்றி அல்ல. சாதாரண அல்லது சாதாரணத்திற்கும் குறைந்த நிலையில் உள்ள மாணவர்களுக்குக் கற்கத் துணைபுரிவதே ஆசிரியரின் நோக்க மாக இருக்கவேண்டும். ஓர் ஆசிரியரின் கற்பிக்கும் முறை, அவரது 'கடைசி மாணவனை' மனத்தில் கொண்டதாயிருக்க வேண்டும். அத்தகைய ஒரு முறையை அவர் கண்டுபிடித்தா ராயின் அந்த முறை எல்லா மாணவர்களுக்கும் உகந்ததாயும் போதியதாயும் இருக்கும் என்பது தெளிவு. இதைக் கருத்தில் கொண்டே 'புரோகிராம் செய்யப்பட்ட கல்வி முறை' (Programmed Instructions) என்ற ஒன்று உருவாகியுள்ளது. இம்

முறையின் தந்தை முன்னர் கூறிய ஸ்கின்னர் ஆவார். இவரது முறைப்படி எந்த ஒரு விஷயத்தையும், சிறு சிறு அம்சங்களாகப் பிரித்துக்கொள்ளவேண்டும். அதாவது ஒரு பெரிய செய்தியைத் துண்டுதுண்டாகச் சிறு அடிப்படைச் செய்திகளாக ஆக்கிக் கொள்ள வேண்டும். இவ்வாறு ஆக்கப்பட்ட துண்டுச் செய்திகளைத் தர்க்க ரீதியாக அடுக்கிக்கொள்ள வேண்டும். ஒவ்வொரு துண்டுச் செய்தியும் ஒன்று அல்லது இரண்டு வாக்கியங்களால் மட்டுமே ஆகியிருக்க வேண்டும். ஒருசமயத்தில் மாணவனுக்கு ஒரே ஒரு துண்டுச் செய்தியைக் கொடுக்கவேண்டும். உடனே அதை அடுத்து அவன் அச்செய்தியை மனத்தில் ஏற்றுக் கொண்டானு, அதாவது கற்றுக்கொண்டானு என்பதைக் கண்டு பிடிக்க ஒரு கேள்விமூலம் கேட்டு அறிந்துகொள்ளலாம். நாம் கேட்கும் கேள்விக்கு அவன் சரியாகப் பதில் கூறினுல் அவனுக்கு அடுத்த துண்டுச் செய்தியை அறிவிக்கலாம். மாணவனது பதில் திருப்திகரமாக இல்லையாயின் முதல் துண்டுச் செய்தியை மீண்டும் அவன் கற்குமாறு செய்யலாம். இம்முறையின் முக்கிய அம்சம், எந்தச் சிக்கல் வாய்ந்த சமாச்சாரத்தையும் துண்டுதுண்டாக உடைத்துச் சிறுசிறு செய்திகளின் தொடராக அடுக்கும்போது மிகவும் பிற்பட்ட மாணவன்கூட இச்சமாச்சாரத்தைப் புரிந்து கொள்ளுமாறு செய்ய இயலும் என்பதே. மேற்கண்ட 'புரோகிராமித்த கல்வி' முறையை அடிப்படையாகக் கொண்டே 'கற்பிக்கும் பொறிகள்' (Teaching Machines) வடிவமைக்கப்பட்டுள்ளன. இத்தகைய பொறிகளை முதலில் கருத்தில் கொண்டு உருவாக்கியவர்களில் ஆர்தர் எல். ரன்யன் (Arthur L. Runyan - 1918), ஸிட்னி எல். பிரஸ்ஸே (Sydney L. Pressey - 1920) ஆகியவர்கள் முதன்மையானவர்கள். இருப்பினும் 1954ஆம் ஆண்டு டாக்டர் ஸ்கின்னர் கற்பிக்கும் பொறிகளின் பயன்களைப் பற்றி தமது 'The Science of Learning and the Art of Teaching' என்ற கட்டுரையில் விவரித்த பின்னரே இப்பொறிகளைப் பற்றி எல்லாரும் சிந்திக்கத் தலைப்பட்டார்கள். இப்பொறி ஒருசமயத்தில் ஒரேஒரு துண்டுச் செய்தியை மட்டும் மாணவனுக்குக் கூறுகிறது. பின்னர் அந்தச் செய்தி சம்பந்தப்பட்ட கேள்வி ஒன்றைக் கேட்கிறது. மாணவன் அந்தக் கேள்விக்கான விடையை அப் பொறியில் 'டைப்' செய்தவுடன் அவனது பதில் சரியானதாக இருப்பின், அடுத்த துண்டுச் செய்தியை மாணவனுக்குக் கொடுக்கிறது. அவனது பதில் ஏற்புடையதாக இல்லையாயின் மீண்டும் முதல் துண்டுச் செய்தியை அவனுக்கு நினைவுறுத்தி, கேள்வியை மீண்டும் கேட்கிறது. முதல் துண்டுச் செய்தியை அவன் புரிந்துகொண்டு, சரிவர விடையிறுத்தாலன்றிக் கற்பிக்கும் கம்ப்யூட்டர் அடுத்த செய்திக்குப் போகாது.

இத்தகைய ஒரு கல்வி முறையின் நன்மைகள் எளிதில் புலனுகும். முதலாவது இம் முறையில் ஒவ்வொரு மாணவனும் கற்கும் செயலில் தனி மாணவனுகச் செயல்படுகிருன். அவனது கற்கும் பொறியும் சேர்ந்து ஒரு மாணவனுக்கு ஓர் ஆசிரியர் என்பது போன்ற ஒரு நிலை ஏற்பட்டு வருகிறது. அதனுல் கற்கும் மாணவன் மற்ற மாணவர்களைப்பற்றிக் கவலைப்படாமல், தனக் குரிய வேகத்தில் கற்றுக்கொண்டுபோக வசதி ஏற்படுகிறது. அடுத்தது இத்தகைய கல்விமுறை ஆசிரியர்களின் விருப்பு வெறுப்புகளுக்கு இடமளிப்பதில்லை. அரசியல், சமூக நிலைகள் பற்றி ஓர் ஆசிரியர் கொண்டுள்ள கருத்துகள் கட்டாயம் ஏதா வது ஓர் உருவில் மாணவர்களை பாதிக்கும். அவர்கள் கற்கும் விஷயங்கள் ஆசிரியரின் கருத்துக்கேற்பத் திரிக்கப்படலாம். ஆனுல் கம்ப்யூட்டரைக்கொண்டு கற்கும் கல்வியில் இத்தகைய அபாயத்திற்கு இடமில்லை. மூன்ருவதாக ஓர் ஆசிரியர் ஒரு குறிப்பிட்ட தலைப்பில் நினைவில் வைத்துக்கொள்ளக்கூடிய செய் தியின் அளவு மிகவும் குறைவாகவே இருக்கமுடியும். ஆனுல் கற்பிக்கும் பொறிகள் இத்தகைய குறைகளால் பாதிக்கப்படுவ தில்லை. பொறியின் நினைவகத்தை வேண்டிய அளவு பெருக்கி, எவ்வளவு செய்திகளை வேண்டுமானுலும் அதன் நினைவகத்தில் இடலாம். புரட்சிகரமான இந்த முறையில் குறைகளும் இல்லா மல் இல்லை. இந்த முறைப்படி ஏராளமான கற்பிக்கும் பொறிகள் தேவைப்படும். அதனுல் ஆகும் செலவு பெருஞ் செலவாகும். பிற்காலத்தில் எலக்ட்ரான் சாதனங்களின் விலை பல மடங்கு குறைந்த பிறகு ஒருவேளை ஒவ்வொரு பள்ளியிலும் நூற்றுக் கணக்கில்கூடக் கற்பிக்கும் பொறிகள் செயல்படலாம்.

'புரோகிராமித்த கல்வி முறையை' எல்லோரும் ஏற்றுக் கொண்டார்கள் என்றும் சொல்லமுடியாது. அதை ஏற்றுக் கொள்ள மறுப்போரின் முக்கிய வாதம், இந்த முறை, கற்றுக் கொள்ளப்போகும் விஷயத்தின் முழுமையான உருவைக் குலைத்து விடுகிறது என்பதும், இம் முறையில் கற்போர் சிறு துண்டு விஷயங்களைக் கற்றுக்கொண்டபோதும், அவையெல்லாம் ஒன்று சேர்ந்த முழுமையான செய்தியைப் பெரும்பாலும் கற்றுக்கொள் ளாமல் மறந்துவிடுகிருர்கள் என்பதாம். எனினும் இத்தகைய எதிர்ப்புகள் காலப்போக்கில் குறைந்துபோகும் என்று எதிர் பார்க்கலாம்.

கற்பிக்கும் முறையில் மட்டுமல்லாமல் தேர்வுகளுக்கான கேள்விகளைத் தயாரித்து, விடைத் தாள்களைத் திருத்தி மாண வர்களின் நிலைகளையும் (Grades) தீர்மானிக்குமாறு கம்ப்யூட் டர்களை ஏவலாம். இந்தத் துறையில் ஏற்கெனவே இங்கிலாந்தில்

பெருமளவுக்குக் கம்ப்யூட்டரைப் பயன்படுத்த ஆரம்பித்து விட்டார்கள். இவைப்பற்றிச் சற்று தெரிந்துக் கொள்ளுவோம்.

தமிழ் நாட்டில் பல்லாயிரக் கணக்கான பள்ளிகள் உள்ளன. நூற்றுக் கணக்கான கல்லூரிகள் உள்ளன. இவற்றுள் பல வேறு பாடங்களும் போதிக்கப் படுகின்றன. முதலில் ஒவ்வொரு பாடத்தையும் (வரலாறு, கணிதம், விஞ்ஞானம் முதலியவை) பல அத்தியாயங்களாகப் பிரித்துக் கொள்ளுவோம். ஒவ்வொரு அத்தியாயமும் ஒரு தனிப்பட்ட பகுதியை கொண்டதாயிருக்கும். எடுத்துக் காட்டாக அத்தியாயம் 8 அக்பரின் ஆட்சியைப் பற்றியும் அத்தியாயம் 9 ஜஹாங்கீரின் காலத்தைப் பற்றியும் இருக்கலாம். இதன் பின்னர் ஒவ்வொரு அத்தியாயத்திலும் கற்பிக்கப் பட்டுள்ள செய்திகளுக்கான கேள்விகளைத் தயாரிக்க லாம். இத்தகைய கேள்விகளை, கல்லூரி நிலை, பள்ளி நிலை இவற்றிற்கு ஏற்றவாறும் பாடம், அத்தியாயம் வாரியாகவும் தயாரித்து உரிய குறியீட்டு எண்களைக் கொடுத்து ஒரு மைய இடத்தில் (திருச்சிராப்பள்ளி என்று வைத்துக் கொள்ளுவோம்) உள்ள பெரிய கம்ப்யூட்டர் ஒருங்கில் வைக்க வேண்டும்.

இத்தகைய அமைப்பை இங்கிலாந்தில் ஐடம் பேங்க் (Item Bank) என்று கூறுகின்றனர். அவ்விடத்துடன் பலவேறு கல்வி நிலையங்களுக்கும் தொடர்பு இருக்கவேண்டும். திடீரென செங்கல்பட்டு மாவட்டத்தில் உள்ள ஆசிரியர் ஒருவர் அக்பரது ஆட்சியைப் பற்றித் தமது மாணவர்களுக்கு ஓர் தேர்வு நடத் துவது என்று தீர்மானித்தால் அவர், திருச்சிராப்பள்ளி கம்ப் யூட்டர் நிலையத்திற்கு

S 06 HI 0805.

என்ற குறியீட்டை அனுப்பலாம். மேற்கண்ட குறியீட்டில் S என்பது பள்ளி (School) என்பதையும் (கல்லூரியிலிருந்து தனிப்படுத்துவதற்காக) அடுத்துள்ள 06 என்ற எண் 6ம் படி வத்தையும் HI என்பது 'வரலாறு' (History) என்பதையும் 08 என்பது எட்டாம் அத்தியாயம் என்பதையும் 05 என்பது எத்தனை கேள்விகள் வேண்டுமென்பதையும் குறிக்கின்றன. உடனே கம்ப்யூட்டர் பள்ளிக்கூடப்பாட திட்டத்தில் 6ம் வகுப்பு வரலாறு பாடத்தில் 8-வது அத்தியாயத்தில் (அக்பரது காலம்) உள்ள கேள்விகளுள் ஏதேனும் 5 கேள்விகளை அச்சிட்டு ஒரு கேள்வித்தாளே தயாரிக்கிறது. செங்கல்பட்டு பள்ளிக்கூடம் இந்தக் கம்ப்யூட்டர் நிலையத்தோடு டெலக்ஸ் (Telex) மூலம் தொடர்பு கொண்டு இருப்பின் இக்கேள்வித்தாள் உடனடியாக

செங்கல்பட்டு பள்ளியில் உள்ள டெலக்ஸ் பொறியில் இக்கேள்வி கள் அச்சிடப்படுகின்றன.

இம்முறை திறம்பட இயங்க வேண்டுமானால் பலவேறு அடிப் படை விஷயங்களில் பள்ளிகளில் ஒருமைப்பாடு இருக்கவேண் டும். மற்ற யாவற்றையும்விட பாடத்திட்டம், தரம் ஆகிய இரண்டும் எல்லா பள்ளிகளுக்கும் ஒன்றாயிருத்தல் அவசியம்.

மேற்குறித்த முறையை விரிவு படுத்தி கம்ப்யூட்டரையே அது தயாரித்த கேள்வித்தாளுக்கு ஒரு விடைத்தாளும் தயா ரித்துக் கொடுக்கச் சொல்லலாம்; அல்லது மாணவர்களது விடைத் தாள்களைத் திருத்தி மதிப்பிட்டு மாணவர்களுக்கு மதிப் பெண்களைக் கொடுக்குமாறும் செய்யலாம்.

பள்ளிகளுக்கும், கல்லூரிகளுக்கும் பாடதிட்டத்தை வகுப் பதிலும் கம்ப்யூட்டர் பெருமளவுக்கு உதவி செய்யலாம். இன் றைய நிலையில் பாட திட்டங்களை வகுப்பதில் பலவேறு குறை கள் உள்ளன. ஒரே தலைப்பை பல பாடங்களில் படிப்பது சில தலைப்புகள் முற்றிலுமாக விட்டு விடுவது, அல்லது தலைப்புகளை தர்க்கரீதியாக அமைக்காது விடுதல் ஆகியவையாம்.

இவை மட்டுமல்லாது, நாள்தோறும் நடக்க வேண்டிய பள்ளி நிர்வாகம் சம்பந்தப்பட்ட விவரங்களையும், பள்ளிக்கான கால அட்டவணைகளைத் (Time Table) தயாரித்தல் போன்ற வற்றிலும் கம்ப்யூட்டரைத் திறம்படப் பயன்படுத்தலாம்.

பாதுகாப்பு :—

பாதுகாப்புத் துறையில் கம்ப்யூட்டரின் பயன் அளவிடற் கரியது. எதிர்காலத்தின் போர் முறைகளும், படைக் களங்களும் கம்ப்யூட்டர்களின் ஆளுகைக்குட்பட்டு இயங்கப் போகின்றன. அப்போது அணுகுண்டோ, அல்லது கண்டம் விட்டு கண்டம் பாயும் ஏவு கணைகளோ மிகப் பெரும் போர் ஆயுதங்களாகக் கருதப்படமாட்டா. மாறாக கம்ப்யூட்டர்களே மிகப் பயங்கர போர் ஆயுதங்களாகக் கருதப்படும். போரிடும் நாடுகளில் எந்த தரப்பில் மிகவும் வலிமை மிக்க கம்ப்யூட்டர் உள்ளதோ, எந்த தரப்பில் கம்ப்யூட்டரைக் கையாள்வோர் திறமையுள்ளவர்களாக உள் ளனரோ அந்த தரப்பே வெல்லும். மேலும் எதிர்காலப் போர்கள் (உலகப் போர்கள்) முதல், இரண்டாவது உலகப் போர்களைப் போல ஆண்டு கணக்கில் நடக்க போவதுமில்லை. எல்லாம் ஒன்று அல்லது இரண்டு நாட்களில் முடிந்து போய்விடக்கூடும். வலிமை மிக்க ஆயுதங்களையும், ஒரே மூச்சில் பல லட்சகணக்கான

உயிரனங்களை அழித்து, பல்லாயிர சதுர மைல்கள் நிலப்பரப்பைத் தீய்த்தெடுக்கும். பயங்கரப் போர்க்கருவிகள் நிறைந்த இன்று, மிகக் குறுகிய காலத்தில் மிகத் திறமையாகப் படைக்கலன்களைப் பயன்படுத்தி, எதிரிகளின் இயக்கங்களை நிலைகுலையச் செய்வாரே வெல்வர். இதற்கு, ஒவ்வொரு விநாடியும் எதிரியின் படை நிலைகள் வலிமை, படைக் கலங்களின் விவரம் போன்றவைகளை அறிந்து கொண்டு உடனடியாக பதில் நடவடிக்கை எடுக்க வல்ல அமைப்பு ஒன்று மிகவும் இன்றியமையாதது. அத்தகைய ஒரு அமைப்பு மிகை வேகமும், பரந்த நினைவகமும் கொண்ட பெரிய கம்ப்யூட்டர் ஒருங்கை மையமாகக் கொண்டே இயங்க முடியும்.

கம்ப்யூட்டர்களைப் பாதுகாப்பு துறையில் பெருமளவுக்கு அமெரிக்கா பயன்படுத்திக் கொண்டு வருகிறது. (சோவியத் யூனியன் எந்த அளவுக்கு தனது பாதுகாப்புத் துறைகளைக் கம்ப்யூட்டர் மயமாக்கியுள்ளது என்பது பற்றிய விவரங்கள் நமக்குக் கிடைக்கவில்லை). சாதாரணமாக, கடற்படை ஆகாய விமான தரைப்படைகள் மூன்றும் முப்பெரும் பிரிவுகளாக இயங்கும். அவற்றிற்கிடையே ஒத்துழைப்பு இருப்பினும், பெருமளவுக்கு ஒன்றின் செயல்பாடுகள் மற்றவற்றிற்கு முற்றிலும் தெரியாமல் இருக்கும். இக்காரணத்தால் அடிக்கடி இரட்டிப்பு வேலைகள் (Duplication) நடைபெறுவதுடன், ஒரு பிரிவினர் பெற்ற அனுபவத்தின் பயன் மற்ற பிரிவினருக்குத் தாராளமாகக் கிடைக்காமலும் போகலாம். இக்காரணங்களாலும், இன்னும் பிற இத்தகைய காரணங்களாலும், படைகளின் முப்பெரும் பிரிவுகளும் ஒருங்கிணைந்து செயல்படுதல் மிகவும் இன்றியமையாததாய் உள்ளது. இதை முன்னிட்டு அமெரிக்க ஐக்கிய நாடு (U. S.) நேஷனல் மிலிடரி கமாண்ட் சிஸ்டம் (National Miltary Command System–NMCS) என்ற தலைமைக் கம்ப்யூட்டர் நிலையம் ஒன்றை நிறுவ ஏற்பாடாகியுள்ளது. இவ்வேற்பாட்டின்படி முப்படைகளையும் சேர்ந்த 40 கம்ப்யூட்டர்கள் (பலவேறு ரகத்தைச் சேர்ந்தவை). ஒருங்கிணைக்கப்படும். இத்தகைய அமைப்பின் பயனாக, அமெரிக்க ஐக்கிய நாட்டின் முப்படை தலைவர்களும், மற்ற உயர் இராணுவ அதிகாரிகளும், எந்த ஒரு நிமிடத்திலும் உலகம் முழுதும் வியாபித்து இருக்கும் அமெரிக்கப் படைகளில் எந்தெந்தப் பிரிவைச் சேர்ந்த போர் வீரர்கள், உலகின் எந்தெந்த பகுதியில் எவ்வளவு பேர் இருக்கிறார்கள், எத்தகைய படைக் கலங்கள் எங்கெங்கு எவ்வளவு உள்ளன, எவற்றை எங்கிருந்து எங்கு எளிதில் குறுகிய காலத்தில் மாற்ற இயலும் என்பது போன்ற விவரங்களை உடனுக்குடனாக அறிந்து கொள்ள முடியும்.

இத்தகைய பெரும் கம்ப்யூட்டர் நிலையம் முப்பெரும் படைகளை மட்டும் இணைப்பதில்லை. நாட்டின் பாதுகாப்புடன் தொடர்பு கொண்ட மற்ற துறைகளையும் ஸ்டேட் டிபார்ட்மெண்ட், உள் துறைப் பாதுகாப்பு, ஒற்றறியும் துறை ஆகிய எல்லா துறைகளையும் கூட ஒன்றிணைக்கிறது. இத்தகைய மாபெரும் இணைப்பு கூட இன்றைய தேவைக்குப் போதாது என்று அமெரிக்க படைத் தலைவர்கள் சிலரது கருத்தாகும். அமெரிக்க ஐக்கிய நாட்டின் படைத்துறைகளின் விவரங்கள் மட்டும் இன்றி, அமெரிக்காவுடன் பாதுகாப்பு ஒப்பந்தம் கொண்டுள்ள, நேட்டோ, போன்ற ஸ்தாபனத்துடன் தொடர்பு கொண்ட எல்லா நாட்டின் படை நிலவரங்கள் பற்றிய விவரங்களையும் ஒருங்கிணைத்து இன்னும் பெரிய மத்திய நிலையம் ஒன்று வேண்டும் என்று கூறப்படுகிறது; இத்தகைய ஒரு அமைப்பு அமெரிக்காவிற்கும் அதனுடன் போர் ஒப்பந்தம் செய்துக்கொண்ட நாடுகளுக்கும் மிகப் பெரும் அளவில் பாதுகாப்பு அளிக்கும் என்று கூறப்படுகிறது. அமெரிக்காவோ அல்லது அதனுடன் ஒப்பந்தம் செய்துகொண்ட ஒரு நாடோ திடீரென ஏவுகணைகளைக் கொண்டு தாக்கப்படுமானால், உடனே திருப்பித் தாக்க வேண்டியிருக்கும். இத்தகைய எதிர்த் தாக்கு பெருமளவிலும், முற்றிலுமாகவும் (Total) இருந்தாக வேண்டும். ஆனால் இத்தகைய ஒரு தாக்குதலுக்கு கிடைக்கக் கூடிய நேரமோ மிகச் சில நிமிடங்களே. இச்சில நிமிடங்களில் பல்லாயிரக் கணக்கான படை நிலையங்கள் ஆணைகளை அனுப்பித் தாக்குதலை ஆரம்பிக்க வேண்டும். இத்தகைய ஒரு பணிக்கு அளவற்ற வேகமும், பெரிய, பெரிய நினைவகங்களையும் கொண்ட கம்ப்யூட்டர்களே ஏற்றவை.

இத்தகைய ஒரு பெரிய கம்ப்யூட்டர் குடும்பத்தை உருவாக்கும் முயற்சி 1960-ஆம் ஆண்டு ஆரம்பித்தது. ஆரம்ப காலத்தில் பலவேறு இடையூறுகள் ஏற்பட்டன. அவற்றுள் முதன்மையானது மொழிப் பிரச்சினைகளும் வேறு சில பொருந்தாமைகளும் (incompatibilities) ஆகும். உருவாக்க வேண்டிய கம்ப்யூட்டர் குடும்பத்தின் அங்கத்தினர்கள் பலவேறு கம்பெனிகளால் வடிவமைத்து உருவாக்கப்பட்ட பலதிறப்பட்ட கம்ப்யூட்டர்களாம்.

இவற்றை ஒருமிக்க வைக்க பலவேறு இணைப்பு முகங்கள் (Inter face) உருவாக்க வேண்டியிருந்தது. பல இடையூறுகளுக்கு மிடையே உருவான இத்தகைய கம்ப்யூட்டர் குடும்பம் NORAD (NORTH AMERICA AIR DEFENCE COMMAND) ஆகும். இதோடு இணைக்கப்பட்டுள்ள பல அங்கங்களுள் SAGE

(Strategic Air Ground Environment) முக்கியமான ஒன்றுகும். SAGE எவ்வாறு வேலை செய்கிறது என்பதைச் சற்றுக் காண்போம்.

வட அமெரிக்காவில், அமெரிக்க ஐக்கிய நாடுகளையும் கனடாவையும் சேர்ந்த நிலப்பரப்பை பல பகுதிகளாகப் பிரித்துள்ளார்கள். ஒவ்வொரு பகுதியைச் சேர்ந்த வானப்பகுதியையும் ஒரு ராடார் தொடர்பின் கண்காணிப்பில் வைக்கிறுர்கள். ஒரு குறிப்பிட்ட பகுதியின் ராடார் தொடர், அந்த பகுதியின் வான் பரப்பை முற்றும் கண்காணிக்கிறது. எடுத்துக் காட்டாக அலாஸ்கா பகுதியில் ஒரு குறிப்பிட்ட பகுதியில், ஒரு திசையிலிருந்து விமானம் ஒன்று வருவதாகக் கொள்ளுவோம். உடனே அங்குள்ள ராடார் நிலையங்கள், அந்த விமானத்தைப் பற்றிய விவரங்களை கம்ப்யூட்டர் நிலையத்திற்கு அனப்புகின்றன.

கம்ப்யூட்டரின் நினைவகத்தில் ஒவ்வொரு பகுதியைச் சேர்ந்த விண்வெளிக்கான வரைப்பட விவரம் ஒன்று உள்ளது. அப்பகுதியில் செல்லக் கூடிய விமானங்களின் (பயண விமானங்கள் போன்றவை) முழு விவரங்களும் நேரம், திசை, அமைப்பு போன்றவையும் அடங்கியிருக்கிறது. வந்து கொண்டிருக்கும் விமானம் பற்றிய விவரங்களைப் பெற்றக் கம்ப்யூட்டர், உடனே அந்த குறிப்பிட்ட நேரத்தில் அந்தக் குறிப்பிட்ட திசையிலிருந்து விமானம் ஒன்று எதிர்பார்க்கப்பட்ட பயணி விமானம் தானு என்று விமான வரைப்படத்துடன் ஒப்பு நோக்கி பரிசோதிக்கிறது. அது எதிர்பார்க்கப்பட்ட விமானமாயின் அதை புறக்கணித்து விடுகிறது. அதற்கு மாறுக, வந்து கொண்டிருக்கும் விமானம் அந்த நேரத்தில் எதிர் பார்க்கபடாத ஒன்று என்று வைத்துக் கொள்ளுவோம். உடனே தரைக் கண்காணிப்பு நிலையத்திற்குக் கம்ப்யூட்டர் செய்தி அனுப்புகிறது. இச் செய்தியில் வந்து கொண்டிருக்கும் விமானத்தைப் பற்றிய முழு விவரங்களும் தரை கண்காணிப்பு நிலையத்திற்கு அளிக்கப்படுகிறது.

இந்த நிலையில் அடுத்தாற் போல் என்ன செய்ய வேண்டும் என்று நிர்ணயிப்பது கண் காணிப்பு நிலையத்தில் உள்ள மனிதர்களே. இவர்களுகளு பல திறப்பட்ட முறைகளில் செயல் படுவதற்கான வசதி உள்ளது. உதாரணமாக வந்து கொண்டிருக்கும் விமானத்தைப் புறக்கணித்து விடு மாறு கம்ப்யூட்டரை அவர்கள் ஆணையிடலாம். அல்லது அவ்விமானத்தைப் பற்றி அறிந்து கொள்ள ஓரிரு போர் விமானங்களை அனுப்ப முடிவு

செய்யலாம். அத்தகைய முடிவு எடுக்கும் பட்சத்தில் இந்த போர் விமானிகளுக்குத் தேவையான புள்ளி விவரங்களை யாவற்றையும் கொடுத்து, அவர்களை, வந்துக் கொண்டு இருக்கும் விமானத்தின் மிக அருகே வரை கம்ப்யூட்டர் வழி நடத்திச் செல்லுகிறது. அதன் பின்னர், கம்ப்யூட்டர் ஒதுங்கிக் கொள்ளுகிறது. பகை விமானத்தைக் கட்டாயப்படுத்தி இறங்க சொல்லுவதா அல்லது சுட்டு வீழ்த்துவதா என்பன போன்ற முடிவுகளை போர் விமானத்தில் செல்லும் விமானிகள்பால் விட்டு விடுகிறது. இதை விடுத்து, கம்ப்யூட்டர் 'பகை விமானம் ஒன்று வருகிறது' என்று அறிவித்தவுடன் தரைக் கண்காணிப்பு நிலையத்தார் கம்ப்யூட்டரையே உரிய நடவடிக்கை எடுக்குமாறு கூறி விடுகிறார்கள் என்று வைத்துக் கொள்ளுவோம். உடனே, கம்ப்யூட்டர் தனது கட்டுப்பாட்டில் உள்ள பல ஏவுகணைத்தளங்களுள் உரியனவற்றைத் தேர்ந்தெடுக்கிறது. இந்த தளங்களில் மினிட்மேன் கணைகள் (Minuteman) நிலையத்திற்கடியில் வைக்கப்பட்டு உறுதியான எஃகு பாலங்களால் மூடப்பட்டுள்ளன. கம்ப்யூட்டரின் ஆணையின் கீழ் இந்த எஃகு கதவுகள் தாமாகத் திறந்து கொள்ளுகின்றன. உடனே ஏவுகணையும், உரிய கோணத்தில் நிறுத்தப்பட்டு புறப்படுவதற்கான ஆயத்தங்கள் செய்யப்படுகின்றன. இந்த ஆயத்தங்கள் முடிவு பெற்ற உடன் கம்ப்யூட்டர் இந்த கணையை தானே செலுத்துகிறது; செலுத்திய பின்னர் இந்தக் கணையை, தொடர்ந்து, வந்து கொண்டிருக்கும் விமானத்தை நோக்கி வழி நடத்திக் கொண்டே போய் விமானத்தின் மிக அருகே போனதும் விட்டுவிடுகிறது. அப்போது ஏவுகணைகளிலிருக்கும் வெப்ப மோப்பிகளின் (Heat Sensors) உதவியால் ஏவுகணை தானாகவே விமானத்தைத் துரத்திச் சென்று சுட்டு வீழ்த்துகிறது. விமானத்தைக் கண்டதிலிருந்து அதைச் சுட்டு வீழ்த்துவது வரை கம்ப்யூட்டர் எடுத்துக் கொள்ளும் நேரம் இரண்டிலிருந்து மூன்று நிமிடங்களேயாம்.

இத்தகைய வான்பாதுகாப்பு (Air Vigilance) மட்டுமல்லாமல் நோரேட் (NORAD) மேலும் பல வேலைகளையும் செய்கிறது. நோரேட்டின் கட்டுப் பாட்டில் இயங்கும் 'ஸ்பாடேட்ஸ்' கம்ப்யூட்டர் ஒருங்கு (SPADATS Space Detection and Tracking system) ஒரு பில்கோ - 212 (Philco-212) கம்ப்யூட்டரைக் கொண்டுள்ளது. விநாடிக்கு 1,680,680 கூட்டல்களைச் செய்ய வல்ல இந்த கம்ப்யூட்டருக்கு பல முனைகளிலிருந்தும் — ஏவுகணை முன்னறிவிப்பு ஒருங்கு (Ballistic Missile Early Warning system), கடற்படை வான் காப்பு (Naval space Surveillance)

விண் வெளி தொடர் நிலையம் (space track)— ஆகிய பல வற்றிலிருந்தும் செய்திகள் வந்த வண்ணம் இருக்கும். மலை மலையாக வரும் இச்செய்திகளை ஒன்று சேர்த்து, வகைப்படுத்தி பெரிய அட்டவணை ஒன்றை பில்கோ தயாரித்துக் கொள்ளு கிறது. இந்த அட்டவணையின் உதவியால் எந்த நிமிடத்திலும் விண் வெளியில் எத்தனை கோளங்கள் (செயற்கை) உள்ளன; அவற்றின் பாதைகள் எவை; அவற்றின் பாதையில் ஏற்படக் கூடிய மாறுதல்கள்; நாளையோ, சில நாட்கள் அல்லது மாதங்களுக்குப் பின்னர் எந்த கோளம் விண் வெளி யில் எவ்விடத்தில் இருக்கும் என்பன போன்ற விவரங்களை உடனுக்குடன் அறிவிக்கிறது.

ஸ்பாடாட்ஸ் (SPADATS), நோரெட் (NORAD) ஆகிய வற்றின் சிறப்பம்சம், மலை மலையாக வரும் பலவகை விவரங்களை யும் வகைப்படுத்தி அவற்றைப் பொருளுள்ள வகையில் பயன் படுத்துவதாம். கம்ப்யூட்டர் தவிர வேறு எந்த விதத்திலும் இவ்வளவு செய்திகளை ஏற்று, அவற்றின் அடிப்படையில் முடிவெடுப்பது என்பது கனவிலும் கருத முடியாத ஒன்றும். இவ்வளவு .இருப்பினும் கம்ப்யூட்டரை விரும்பாத அமெரிக்க படைத் தலைவர்களும் இருக்கத்தான் செய்கிறார்கள். அவர்களது வாதம் என்னவெனில், எவ்வளவுதான் கம்ப்யூட்டர் ஆற்ற லுடையதாயிருப்பினும், அதனால் ஒரு வீரனின் போர்திறமையை மதிப்பிடவோ அல்லது ஒரு குறிப்பிட்ட போரின் விளைவு என்ன வாக இருக்கும் என்று முன்கூட்டியே துல்லியமாக கணித்துக் கூறவோ முடியாது என்பதும் ஆகும். அதுவுமல்லாமல் கம்ப் யூட்டரின் கணிப்பை முழுக்க முழுக்க நம்பிப் போர் புரியும்போது போர் நுணுக்க முறையில் தவறுகள் ஏற்பட ஏது இருக்கிறது என்பதும் இவர்கள் வாதம்.

ஆனால் கம்ப்யூட்டரைப் பயன்படுத்த வேண்டும் என்று சொல்லும் கூட்டத்தின் வாதம் என்னவெனில், கம்ப்யூட்டர் தானாக எந்த முடிவையும் எடுக்க போவதில்லை. மிகச் சிக்கலான, கடின சூழ்நிலைகளில் வெற்றிகரமான முடிவுகளை எடுக்கக் கம்ப்யூட்டர் துணைச்செய்கிறது என்பதாம். இன்று இக் கூட்டத் தினரின் கை ஓங்கியிருப்பதால் மேலும் மேலும் பெரும் அளவில் கம்ப்யூட்டர்கள் அமெரிக்கப் பாதுகாப்புத் துறைகளில் பயன் படுத்தப்படும் என்று எதிர்பார்க்கலாம்.

அமெரிக்காவின் ஏதோ ஒரு மூலையில் நிலத்திற்கடியில் ஒரு பெரிய கம்ப்யூட்டர் ஒருங்கு இருக்கிறது. இதற்குப் பாதுகாப்புத்

துறை, 'நாசக் கணிப்பு நிலையம்' (Department of Defence Assessment Centre — Doddac) என்று பெயர் சூட்டியுள்ளது. இக்கம்ப்யூட்டரில், ஒருங்கின் பணிகள் பலதிறப்படும். பகைப் படைகள் சிதறி நிறுத்தப்பட்டுள்ள நிலைகள், அவற்றின் ஆற்றல், அவை கொண்டுள்ள படைக் கலங்கள் ஆகியவற்றைக் கருத்தில் கொண்டு, எதிரி திடீரென்று தாக்கினுல் அதனுல் உடனடியாக அமெரிக்காவிற்கும் அதன் நட்பு நாடுகளுக்கும் எத்தகைய சேதம் விளையும் என்பதைக் கணிக்கிறது.

இத்தகைய கம்ப்யூட்டர்கள் இருப்பதால் அமெரிக்கா எந்த நிமிடமும் போருக்கு தயாராக உள்ளது என்றே கூறலாம்; ஒரு சில நிமிடங்களில், உலகம் முழுதும் போரிட அந்நாடு தனது முப்படைகளையும் தயாராக்கி விடும்.

இங்கு நாம் முக்கியமான ஒன்றை கூறியாக வேண்டும். கம்ப்யூட்டர்களின் இயக்கம் முழுதுமே புரோகிராம்களால் கட்டுப் படுத்தப்படுகின்றன. புரோகிராம் தயாரிப்பவர்கள் எந்த நாட்டை எதிரி நாடு அல்லது எதிரி நாடுகள் என்று கருதி புரோகிராம் தயாரிக்கிருர்களோ, அவையே கம்ப்யூட்டருக்கும் எதிரி நாடாக ஆகிவிடும். ஆனுல் அடிக்கடி அரசியல் திருப் பங்கள் ஏற்படும் இந்நாளில் ஒரு குறிப்பிட்ட நாடே மற்ருெரு நாட்டின் எதிரியாகவோ, அல்லது நண்பனுகவோ இருக்கும் என்று கூற இயலாது. அரசியலில் நிலையான நண்பனோ, நிலை யான எதிரியோ கிடையாது. அமெரிக்காவின் இன்றைய 'எதிரி' யான சோவியத் யூனியன், நாளை அதன் உற்ற தோழனுக ஆகலாம், அல்லது இன்றைய நண்பனுன ஜப்பான், மீண்டும் பெரு விரோதியாக மாறலாம். ஆகையால் மேற் சொன்ன கம்ப்யூட்டர் ஒருங்குகளைக் கட்டுப்படுத்தும் புரோகிராம்கள் அரசியல் மாறுதல்களுக்கு ஏற்ப மாற வேண்டியுமிருக்கும்.

பாதுகாப்பு துறையில் உள்ள கம்ப்யூட்டர்கள் பாதுகாப்பு சம்மந்தப்பட்ட விஷயங்களில் மட்டுமே அக்கறை காட்டு கின்றன என்பதுமில்லை. அமெரிக்கக் கடற்கரைக் காவல் (U. S. Coast Guard) நிறுவனத்தின் AMVER ஒருங்கு (Altantic Merchant vessel Report) ஒரு IBM - 305 கம்ப்யூட்டரைக் கொண்டு உள்ளது. இந்த ஒருங்கு அட்லாண்டிக் சமுத்திரத்தை கடந்து நியூயார்க் துறைமுகத்தின் மூலம் அமெரிக்காவுடன் வர்த்தகத்தில் ஈடுபடும் 55 நாடுகளைச் சேர்ந்த கப்பல்களைக் கண்காணிக்கிறது. 1963-ம் ஆண்டு கிறிஸ்துமஸ் சமயத்தின் போது, 'லக்கோனியா' என்ற கிரேக்கக் கப்பலிடமிருந்து

அபாய அழைப்பு (SOS) வந்தது. கடலில் அதன் இருப்பிடத்தைக் கம்ப்யூட்டருக்குத் தெரிவித்தனர். மறுவினுடி அச்சமயம் அட்லாண்டில் கடலில் இருந்த 850 கப்பல்களில் 'லக்கோனியா' விற்கு மிக அருகில் சென்றுக் கொண்டிருந்த கப்பல் எது என்று கம்ப்யூட்டர் கூற, அக்கப்பலுக்கு உடனே 'லக்கோனியா' வின் பிரயாணிகளைக் காப்பாற்றுமாறு வேண்டுகோள் விடுக்கப்பட்டது.

பாதுகாப்புத் துறையில், போர் புரிவதுமட்டும் சிக்கலான விஷயமல்ல. அமெரிக்கா போன்ற நாட்டின் படைகளின் பல பிரிவுகள் உலகின் பல பகுதிகளிலும் சிதறியிருக்கின்றன. இப்படைகளின் தேவைகள்,— உணவு, உடை, படைக்கலன்கள், உதிரிப் பகுதிகள் (Spare Parts) ஆகியவைகள்—போதிய அளவில் உரிய இடங்களில் சேமிக்கப்பட்டு, வேண்டிய இடங்களுக்கு உடனடியாக அனுப்பப்பட்டாக வேண்டும். அமெரிக்கப் படையில் கிட்டதட்ட 3½ மில்லியன் வகையான பொருள்கள் உள்ளதாகக் கணக்கிடப்பட்டுள்ளது. ஓரிடத்தில் ஒரு பொருள் தேவைப்பட்டதும் அதைத் தொடர்ந்து பல செயல்கள் நிகழ்கின்றன. வேண்டிய பொருளுக்கான தேவை—குறிப்பு (Indent) தயாரித்து அளித்ததும், அருகாமையில் உள்ள பண்டகசாலையிலிருந்து (Stores) அப்பொருள் அளிக்கப்படுகிறது. உடனே பண்டகசாலையில் அப்பொருளின் இருப்புக் கணக்கில் மாற்றம் ஏற்படுகிறது. இம்மாற்றம் உடனே அப்பொருள்களுக்கான புது தேவை உத்தரவு ஒன்றைத் தலைமையகத்திற்கு அனுப்பவேண்டியநிலை ஏற்படலாம். மீண்டும் தலைமையகத்திலிருந்து அப்பொருளை அனுப்பும்போது அதற்கான விவரங்கள் பல நிலைகளில் கணக்கு வைக்கப்பட வேண்டும். இத்தகைய மூன்றரை மில்லியன் வகையான பொருள்களைக் கையாண்டு குழப்பமில்லாமல் கணக்கை வைக்கக் கம்ப்யூட்டரால் மட்டுமே இயலும்.

7 பிற

1 கம்ப்யூட்டர் நிலையம் அமைத்தல்

கம்ப்யூட்டர் நிலையம் ஒன்றை அமைப்பதென்ருல் பலவேறு உண்மைகளையும் கருத்தில் கொள்ளவேண்டும். கம்ப்யூட்டர் நிலையத்திற்கான செலவும், அதனுல் கிடைக்கும் லாபமும் ஒப்பிட்டு நோக்கப்படவேண்டும். அப்படி நோக்கி கம்ப்யூட்டர் நிலையம் ஒன்றை ஏற்படுத்துவது என்று தீர்மானித்தால் கம்ப்யூட்டர் வருவதற்கு முன்னர் பல முன்னேற்பாடுகளைச் செய்தாக வேண்டும்.

அமெரிக்கா போன்ற நாடுகளில்கூட கம்ப்யூட்டர் முதலில் வந்தபொழுது அது ஒரு 'அந்தஸ்து—அடையாளம்' (Status Symbol) எனப் பல கம்பெனிகளால் கருதப்பட்டது. கம்பெனி A-ல் கம்ப்யூட்டர் இருந்து, B-ல் இல்லையாயின் A-ன் அந்தஸ்து B-யினுடையதைவிட உயர்ந்தது என்ற தவருன எண்ணப் போக்கு இருந்தது. இதன் விளைவாகப் பல வணிக நிலையங்கள் (Business Establishments) தங்களது பணிக்குக் கம்ப்யூட்டர்கள் தேவையில்லாமலிருந்துங்கூட அவற்றை வாங்கி வைத்துக் கொண்டு பெரு நஷ்டம் அடைந்தன. ஆனுல் இன்றைய நிலை வேறு.

சிறிய வணிக நிலையங்களிலும், சிறு ஆராய்ச்சிக் கூடங்களினும் கம்ப்யூட்டரின் உதவி தேவைப்படலாம்; ஆனுலும் ஒரு கம்ப்யூட்டருக்கு தேவையான முழு அளவு பணி அவர்களிடம் இருக்காது. இத்தகைய சூழ்நிலையில் விலைமிக்க கம்ப்யூட்டரை வாங்கி வைத்துக்கொண்டு ஒரு நாளைக்கு ஓரிருமணி நேரங்கள் மட்டுமே கம்ப்யூட்டரைப் பயன்படுத்திஞல் அது வீண் விரயமாகும்:

இன்று பல கம்ப்யூட்டர் நிலையங்கள் தங்களது கம்ப்யூட்டர் நிலையத்தின் நேரத்தை விற்கின்றன. ஒருமணிநேர உபயோகத்

திற்கு இவ்வளவு தொகை என்று குறிப்பிட்டு கம்ப்யூட்டரை வாடகைக்கு விடுகிருர்கள். இந்தியாவில் இத்தொகை ரூ. 100 லிருந்து ரூ. 1500 வரை, கம்ப்யூட்டர் அமைப்பின் வேகம், காந்தத்தட்டு, காந்தநாடா போன்ற மற்ற சௌகரியங்களையும் (Facilities) பொறுத்து, மாறுபடுகிறது. சிறு வணிக நிலையங் களும், ஆராய்ச்சி நிலையங்களும் தங்களுக்கு என்று ஒரே ஒரு புரோகிராமரை மட்டும் வைத்துக்கொண்டு, நேரத்தை விற்கும் கம்ப்யூட்டர் நிலையங்களில் வாடிக்கையாளராக இருக்கலாம்.

ஒரு பெரிய அமைப்பின் (establishment) அலுவல்களை கம்ப் யூட்டர் மயமாக்குவது என்பதும் எளிதல்ல. பெரிய அமைப்பு களில் ஒன்றிரண்டு அலுவல்களை மட்டும் முதலில் கம்ப்யூட்டர் அலுவல்களாக மாற்றுவார்கள். அவ்வாறு மாற்றும்போது தமக்கு வேண்டிய கம்ப்யூட்டர் நேரத்தை ஒரு கம்ப்யூட்டர் நிறுவனத் திடமிருந்து விலைக்கு வாங்கிக்கொள்வார்கள். இந்த ஒன்றிரண்டு அலுவல்களும் வெற்றிகரமாக கம்ப்யூட்டர் அலுவல்களாக மாற்றி முடிக்கப்பட்டதும், மேலும் சில அலுவல்களைக் கம்ப்யூட்டர் கணிப்பாக மாற்றுவார்கள். இவ்வாறே சிறிது சிறிதாக கம்ப் யூட்டர் அலுவல்களைப் பெருக்கிக்கொண்டே போய், ஒரு குறிப் பிட்ட நிலை வந்ததும், இந்த அமைப்பு ஒரு கம்ப்யூட்டருக்கு ஆர்டர் செய்யும்; கம்ப்யூட்டர் வந்தவுடன் அதற்கு அளிக்க வேண்டிய கணிப்புகள் யாவும் தயாராக இருக்கும். இத்தகைய ஏற்பாட்டின் காரணமாகப் பெருமளவில் பொருள் நஷ்டம் தவிர்க்கப்படும். கம்ப்யூட்டர் வந்த பின்னரும்கூட, கம்ப்யூட் டரின் முழுநேரமும், அந்த ஸ்தாபனத்திற்கு தேவையில்லை யாயின் மற்ற கம்ப்யூட்டர் நிலையங்களைப் போலவே இந்த ஸ்தாபனமும், மற்றோருக்கு கம்ப்யூட்டர் நேரத்தை விற்கலாம்.

ஒரு பணியை மனிதர்களின் கையிலிருந்து கம்ப்யூட்டருக்கு மாற்றும்போது சிறிது காலத்திற்கு அப்பணி, மனிதர்களாலும், கம்ப்யூட்டராலும் தொடர்ந்து இணையாக (parallel) செய்யப்பட வேண்டும். அப்போது இரண்டு முறைகளையும் ஒப்புநோக்கி, கம்ப்யூட்டர் முறையில் ஏதேனும் தவறு ஏற்படுகிறதா என் பதைக் கண்டறிய ஏதுவாகும். சிலகாலம், கணிப்பைக் கையா லும் கம்ப்யூட்டராலும் இணையாகச்செய்து, கம்ப்யூட்டர் கணிப்பு திருப்திகரமாக அமைந்த பின்னர் கைக் கணிப்பை விட்டுவிட லாம். இதுவே சாதாரணமாக கைக்கொள்ளப்படும் முறையாம்.

பெரும்பாலான நிறுவனங்கள் இன்று தங்கள் சென்டர்களுக் கான கம்ப்யூட்டர்களை சொந்தமாக வாங்கிக்கொள்வதில்லை. கம்ப்யூட்டர் உற்பத்தியாளர்களிடம் பலவிதமான பொறிகளையும்

மாத வாடகைக்கு வாங்கிக்கொள்ளுகிருர்கள். இதில் பலவித மான சௌகரியங்களும் உண்டு. மாத வாடகை ஏற்பாட்டின்படி வாங்கப்படும் கம்ப்யூட்டர்களில் பழுதுஏற்படாமல் காப்பதோ, பழுது ஏற்படின் உடனடியாக உரிய நடவடிக்கைகளை மேற் கொண்டு பழுது பார்ப்பதோ கப்ப்யூட்டரை வாடகைக்கு கொடுக்கும் கம்பெனியின் பொறுப்பாகும். கம்ப்யூட்டரை வாட கைக்கு வாங்குவோர், கம்ப்யூட்டர் பழுதுபட்டு செயல்படாமல் இருந்தால், செயல்படாமல் இருக்கும் காலத்திற்குரிய தொகையை வாடகையிலிருந்து கழித்துக்கொள்வார்கள்; எனவே பழுது ஏற் படின் காலதாமதமின்றி, கம்ப்யூட்டரை வாடகைக்கு கொடுப் போர் அதைச் சரிசெய்துவிடுகின்றனர்.

மேலும் புதுமையான கம்ப்யூட்டர் ஒன்று உற்பத்தி செய்யப் பட்டு சந்தைக்கு வரும்போது நம்மிடம் கைவசமுள்ள பழைய வாடகைக் கம்ப்யூட்டரை எடுத்துக்கொண்டு புதிய பொறியை நமக்குக் கொடுக்குமாறு கேட்கலாம். ஆனல் கம்ப்யூட்டரைச் சொந்தமாக வாங்கி வைத்திருப்பின் இத்தகைய மாற்றம் செய்ய வேண்டுமாயின் அதனல் குறிப்பிடத்தக்க பண நஷ்டம் ஏற்பட லாம். கையிலுள்ள பழைய கம்ப்யூட்டரை விற்றுவிட்டும் பதி லாக புதிய ஒன்றை வாங்குவதற்கு, முதலில் நமது பழைய பொறியை விற்கவேண்டும். இதற்கு மேற்கொள்ளப்படவேண் டிய சிரமங்களும் பண விரயமும் மிக அதிகம்.

2 கம்ப்யூட்டர் நிலையப் பணியாளர்

கம்ப்யூட்டர் நிலையங்களில் பணிபுரிவோர் பலதிறப்பட்ட நிலைகளில் உள்ளனர். அவர்களைக் கீழ்க்காணும் முறையில் வரிசைப்படுத்தலாம்.

1. சுத்திகரிப்பாளர் (Cleaner)
2. யூனிட் ரெகார்டு அறை உதவியாளர் (Unit Record Room Assistants)
3. யூனிட் ரெகார்டு அறைக் கணிப்பாளர் (Unit Record Room Supervisor)
4. கம்ப்யூட்டர் ஆப்பரேடர்
5. புரோகிராமர் (Programmer)
6. சிஸ்டம் அனலிஸ்ட் அல்லது ஒருங்காய்வாளர் (System Analyst)

7. தலைமை புரோகிராமர் அல்லது தலைமை ஒருங்கறிவோர் (Chief Programmer or Chief Systems Analyst)

8. மேனேஜர் அல்லது டைரக்டர் (Manager or Director)

மேற்கூறிய வரிசையில் காணும் அமைப்பிலும், பெயர்களிலும் சிறு சிறு மாறுதல்கள் இருக்கக்கூடும். நிலையத்தின் அளவுக்கு ஏற்ப ஒவ்வொரு வகையிலும் பணியாளர்களின் எண்ணிக்கை மாறக்கூடும்.

கம்ப்யூட்டர் நிலையத்தின் சுத்திகரிப்பாளரின் பணி, அவரது உத்தியோகப் பெயரிலேயே உள்ளது. நிலையத்தில் உள்ள பொறிகளைத் துடைத்துச் சுத்திகரிப்பதும், கம்ப்யூட்டர் அறை, துளைப்பொறி அறை, யூனிட் ரெகார்டு அறை ஆகியவற்றைத் துப்புரவாக வைத்துக்கொள்ளுவதாம்.

யூனிட் ரெகார்டு அறை உதவியாளர்களில் துளைப் பொறி இயக்குவோர் ஒரு பகுதியினராம். இவர்களது பணி, புரோக்ராம்களையும், அடிப்படை விவரங்களையும் துளைப் பொறிகளின் உதவிகொண்டு துளைத்து கம்ப்யூட்டர்களில் பயன்படுத்த ஏதுவாக துளைக் கார்டுகளைத் தயாரிப்பதாம். புரோக்ராம்களும், அடிப்படை விவரங்களும், துளைப் பொறியாளரின் வசதிக்காக, தனிப்பட்ட, கட்டமிட்ட காகிதங்களில் (Coding Sheets) எழுதப்படுகின்றன. இக்காகிதங்களில் உள்ள கட்டங்கள் துளைக் கார்டில் உள்ள 80 இடங்களையும் வரிசைப்படிக் குறிப்பிடுகின்றன. புரோக்ராம் ஆணைகளையும், அடிப்படை விவரங்களையும் இக் குறியீட்டுத் தாள்களில் (Coding Sheets) எழுதும்போது, ஒவ்வொரு குறியீடும் அல்லது எழுத்தும் துளைக்கார்டில் எங்கு துளைக்கப்படவேண்டுமோ அந்த இடத்திற்குரிய கட்டத்தில் புரோக்ராமர் எழுதிக்கொடுக்கிருர். இத்தாளின் உதவியால், துணைப் பொறியாளர் புரோக்ராம் ஆணைகளைத் துளைக்கார்டில் எங்கு ஆரம்பித்து எங்கு முடிக்கவேண்டும்; அடிப்படை விவரங்களை எங்கெங்கு எவ்விதம் துளைக்கவேண்டும் என்பன போன்றவற்றை எளிதில் அறிந்துகொள்ள முடிகிறது.

துளைப்பொறியை இயக்குவோரே பெரும்பாலும் சரிபார்த்தியையும் இயக்குவர். ஒரு கம்ப்யூட்டர் நிலையத்தில் உள்ள துணைப் பொறியாளரது எண்ணிக்கை ஒரேமாதிரியாக இருக்காது. பெரும்பாலும் ஆராய்ச்சி நிலையங்கள், கல்விக்கூடங்கள் ஆகியவற்றில் உள்ள கம்ப்யூட்டர்கள் பெரிய அளவில் விஞ்ஞானக் கணிப்புகளையே (Scientific Computation) செய்கின்றன

அத்தகைய நிலையங்களில் உள்ள துளைப் பொறியாளரின் எண்ணிக்கை மிகவும் குறைவாகவே இருக்கும். அதே சமயத்தில் பெருமளவுக்குத் தொழில்துறை வாணிபக் கணிப்புகள் (Business Computation) நடக்கும். கம்ப்யூட்டர் நிலையங்களில் ஏராளமான கார்டுகள் துளைக்கப்படவேண்டி யிருக்கும். இத்தகைய நிலையங்களில் துளைப் பொறியாளரின் எண்ணிக்கை அதிகமாக இருக்கும். இதையே வேறுவிதமாகக் கூறப்போனால் ஒரு கம்ப்யூட்டர் நிலையத்தில் உள்ள துளைப் பொறியாளரின் எண்ணிக்கை அந்நிலையத்தில் துளைத்துச் சரிபார்க்கப்படவேண்டிய கார்டுகளின் அளவைப் பொருத்தே அமையும்; முற்றிலும் ஒரே மாதிரியான கம்ப்யூட்டர்களைக் கொண்ட இரண்டு நிலையங்களில் ஒன்றில் ஒன்றிரண்டு துளைப் பொறியாளரும் மற்றதில் 20 அல்லது 30 துளைப் பொறியாளரும் இருப்பது சாத்தியமே.

துளைப் பொறியாளர் மட்டுமின்றி, மற்ற பொறிகளையும், சார்ட்டர், கொலேட்டர், போன்ற பொறிகளையும் இயக்குவதற்கு ஆட்கள் தேவை. ஒரு கம்ப்யூட்டர் நிலையத்தில் பெருமளவுக்கு இத்தகைய பொறிகள் பயன்படுத்தப்படுமானால் அதற்கென்று தனி ஆட்கள் இருப்பார்கள். ஆனால் ஓரிரண்டு யூனிட் ரெகார்ட் பொறிகளை மட்டும் வைத்திருக்கும் கம்ப்யூட்டர் நிலையங்களில் துளைப் பொறியாளர்களே இப் பொறிகளை இயக்கவும் செய்வர்.

துளைப் பொறியாளராகப் பயிற்சிப் பெறுவது, தட்டெழுத்துப் பொறிப் பயிற்சி பெறுவதற்கு ஈடானதே. பலவேறு தனிப்பட்டோர் நிலையங்களும், கம்ப்யூட்டர் உற்பத்திசெய்து விற்கும் நிலையங்களும் இத்தகைய பயிற்சியை அளிக்கின்றன. ஒரு நல்ல தேர்ந்த துளைப்பொறியாளர் ஒரு மணிக்கு 175 முதல் 200 முழு கார்டுகள் (80 இடங்கள்) வரை துளைக்க இயலும்.

மற்ற யூனிட் ரெகார்டுப் பொறிகளை இயக்குவதற்கு, அவற்றை விற்கும் கம்பெனியினரே உரிய பயிற்சிகளையும் அளிக்கிறார்கள்.

துளைப் பொறியாளரும், யூனிட் ரெகார்டு பொறியாளர்களும் பெருமளவில் பணிபுரியும் நிலையங்களில் அவர்களது வேலைகளை ஒழுங்குபடுத்தி மேற்பார்வை பார்ப்பதற்கென காணிப்பாளர்கள் (Supervisors) இருப்பார்கள். இவர்கள் பெரும்பாலும், துளைப் பொறிகளிலும், யூனிட் ரெகார்டுப் பொறிகளிலும் அனுபவமுடையவர்களாக இருப்பர். அநேகமாக துளைப் பொறியாளராகவோ, யூனிட் ரெகார்டு பொறியாளராகவோ இருந்து அனுபவம் பெற்றவர்களே பதவி உயர்வு பெற்று இப் பணியைப் பெறுவர்.

அடுத்த நிலையில் உள்ளோர் கம்ப்யூட்டர் ஆபரேடர் ஆவார். இவரது பணி கம்ப்யூட்டரையும் அதோடு இணைந்துள்ள பல திறப்பட்ட பொறிகளையும் இணைத்தியக்குவதும் ஆகும். இந்தப் பணிக்கு கம்ப்யூட்டரை விற்கும் கம்பெனியினரே பயிற்சி அளிக்கிறார்கள். பெரிய கம்ப்யூட்டர் ஒருங்குகளில் ஒரே சமயத்தில் ஒன்றுக்கு மேற்பட்ட ஆபரேடர்கள் தேவைப்படலாம். மேலும் பெரும்பாலான கம்ப்யூட்டர் நிலையங்கள் ஒன்றுக்கு மேற்பட்ட ஷிப்ட் (Shift) முறைகளில் இயங்கும். ஒரு நாளில் 24 மணி நேரமும் இயங்கும் கம்ப்யூட்டர் நிலையங்கள் பல உள்ளன. இத்தகைய நிலையங்களில் அதற்கேற்றாற்போல் ஆப்ரேட்டர்களின் எண்ணிக்கையும் அதிகமாக இருக்கும்.

அடுத்த நிலையில் (Stage) உள்ளோர் புரோக்ராமர் ஆவர். இவரது பணி கம்ப்யூட்டரில் செய்யவேண்டிய கணிப்பிற்கான புரோகிராமர்களைத் தயாரிப்பதாம். கணிப்புகளுக்கான விதி முறைகளையும் (Procedures) கணித்துப் பெறவேண்டியவை (Out put) யாவை என்பதையும் இவரிடம் கூறிவிட்டால், அதற்கு ஏற்ற கட்டளைகளையும், அடிப்படை விவரங்களையும் இவர் தயாரிக்கிறார். சில வகையான கணிப்புகளில் ஒன்றுக்கு மேற்பட்ட புரோகிராமர்கள் ஒன்றுசேர்ந்து ஒரே புரோக்ராமைப் பாகங்களாகப் பிரித்துக்கொண்டு தனித்தனியே எழுதிப் பின்னர் ஒன்றுசேர்ப்பதும் உண்டு. சிக்கல் வாய்ந்த சில புரோக்ராமர்களை எழுதி வெற்றிகரமாகச் செயல்படுத்த மாதக்கணக்கில் ஆகலாம்.

புரோக்ராமர், தனது புரோக்ராமை எழுதி முடித்ததும், அதைத் துளைப் பொறியாளருக்கு ஏற்ற வகையில் குறியீட்டுத் தாள்களில் (Coding Sheets) எழுதுகிறார். இவ்வாறு எழுதப்பட்ட புரோக்ராம், துளைப் பொறியாளருக்கு அனுப்பப்படுகிறது. அதைக்கொண்டு, துளைப் பொறியாளர் புரோக்ராம் துளைக் கார்டுகளைத் தயாரிக்கிறார். இத் துளைக் கார்டுகள் புரோக்ராமருக்கு அவரது கையெழுத்துப் பிரதியுடன் அனுப்பப்படும். அப்போது புரோக்ராமர் துளைக் கார்டுகளில் துளைக்கப்பட்டுள்ளதையும் தாம் எழுதிய புரோகிராமையும் ஒப்புநோக்கி பிழை யேதேனும் இருக்கிறதா எனச் சோதிக்கிறார். ஏதேனும் பிழைபட்ட கார்டுகளுக்குப்பதிலாகப் புதிய கார்டுகள் துளைக்கப்பட்டு திருத்தங்கள் செய்யப்படுகின்றது. இத்தகைய திருத்தங்கள் முடிந்ததும், புரோக்ராம் கம்ப்யூட்டர் அறைக்கு அனுப்பப்படுகிறது. கம்ப்யூட்டர் அறையில் உள்ள ஆப்ரேட்டர் தம்மிடம் வந்த புரோக்ராம் எழுதப்பட்டுள்ள மொழிக்கேற்ற கம்பைலர் அல்லது புராசசரைக்

கம்ப்யூட்டரின் நினைவகத்தில் ஏற்றி, அதன் பின்னர் புரோ க்ராமை இடுகிறார். சில கம்ப்யூட்டர்களில் முன்னமேயே கம்பை லர்கள் ஏற்றப்பட்டிருக்கும். சில கண்ட்ரோல் கார்டுகளைக் கொண்டு வேண்டிய கம்பைலரை உபயோகிக்கலாம். இந்த வெள்ளோட்டத்தில் கணிப்பின் விடை கிடைக்காது. எவ்வளவு தான் கவனமாகப் புரோக்ராமை எழுதியிருப்பினும், எவ்வளவு தான் முயன்று புரோக்ராம் கார்டுகளைக் கவனமாகச் சேர்த்திருப் பினும் பெரிய புரோக்ராம் ஒன்றில் ஓரிரு புரோக்ராம் பிழைகள் (புரோக்ராம் மொழியின் இலக்கணவிதி மீறியதால் ஏற்படக் கூடிய பிழைகள்) ஏற்படுவது இயல்பே. இத்தகைய தவறுகளை நமது கம்பைலர் எடுத்துக்காட்டுகிறது. அநுபவம் மிகுந்த புரோக்ராமர் தனது புரோக்ராமின் வெள்ளோட்டத்தின் முடிவில் எதிர்பார்ப்பது ஒரு பிழை—அட்டவணையே (Error list) அல்லாமல் முடிவான விடையல்ல. இந்தப் பிழை அட்ட வணையும், புரோக்ராம் கார்டுகளும் இப்போது மீண்டும் புரோ க்ராமுக்கே திருப்பி அனுப்பப்படுகின்றன. புரோக்ராமர் தான் பெற்ற பிழை அட்டவணையின் உதவியால் புரோக்ராம் பிழை களைத் திருத்தி மீண்டும் புரோக்ராமைக் கம்ப்யூட்டர் அறைக்கு அனுப்புகிறார். இம்முறை புரோகிராமைக் கம்ப்யூட்டரில் ஓட்டும் போது 'இலக்கணப் பிழைகள்' எதும் இல்லாமல் இருக் கலாம். புரோக்ராம், இலக்கண விதிப்படி சரியாக இருக்கும் போது நமக்குக் கணிப்பின் விடை கிடைக்க வாய்ப்பு உண்டு. ஆனால், இந்த விடை சரியானதாக இருக்கும் என்றுகூற இய லாது. ஏனெனில் புரோக்ராமில் இலக்கணப் பிழைகளைத்தவிரப் பிறிதொருவகைப் பிழைகளும் ஏற்பட ஏதுவுண்டு. எடுத்துக் காட்டாக

$$Y = (A + B) * (C - D) / E$$

என்ற ஆணைக்குப் பதிலாக, தவறுதலாக

$$Y = (A B) * C D / E$$

என்று கார்டு துளைக்கப்பட்டிருந்தால் என்னவாகிறது என்று பார்ப்போம். கம்ப்யூட்டரைப் பொறுத்தவரை மேலே கொடுக் கப்பட்ட இரண்டு ஆணைகளும் நிறைவேற்றப்படத்தக்கவையே. இரண்டாம் ஆணையில் உள்ள C - Dஐ அடைப்புகளில் இட்டிருந் தாலும்சரி, இடாவிடினும்சரி, அது பற்றிக் கம்ப்யூட்டருக்குக் கவலை ஏதும் இல்லை. ஏனெனில் நமது கணிப்பில் பயன்படுத் தப்படும் ஆணைகளில் உள்ள சூத்திரங்கள் முதலியவை சரியான வையா இல்லையா என்பதை நாம்தான் பார்த்துக்கொள்ளவேண் டுமே யல்லாது கம்ப்யூட்டர் அதைச் செய்யுமென்று எதிர்பார்க்க

முடியாது. (எந்த கம்ப்யூட்டரும் இதைச்செய்ய முடியாது) மேற் கண்ட இரண்டு ஆணைகளில் முதலாவதற்குப்பதிலாக இரண் டாவதை துளைக்கார்டில் துளைத்திருந்தால் நமக்கு இறுதியில் 'விடை' கிடைக்கக்கூடும்; ஆனால் அது தவறான விடையாகத் தான் இருக்கும். ஆகவே கம்ப்யூட்டரிலிருந்து வந்த விடைகள் சரியானவைதானா என்று சோதித்தல் மிகவும் அவசியம் என்றா கிறது; இதுவும் புரோக்ராமரது பொறுப்பாம்.

ஒரு கம்ப்யூட்டர் நிலையத்தில் கம்ப்யூட்டருடன் நேரடித் தொடர்பில்லாத, யூனிட் ரெகார்டு பொறிகள் பல இருக்கலாம். ஒரு குறிப்பிட்ட கணிப்பைச் செய்ய என்ன முறை பின்பற்றப்பட வேண்டும், எந்தெந்த யூனிட் ரெக்கார்டு பொறிகளைப் பயன் படுத்தலாம், எந்த அளவுக்குக் கம்ப்யூட்டரின் பணி தேவைப் படும் என்பன போன்றவற்றை முதலிலேயே தீர்மானம் செய் வது மிகவும் இன்றியமையாதது. கம்ப்யூட்டருடன் தொடர்பு டைய ஒவ்வொரு அம்சமும் பணச்செலவு மிக்கதாயிருப்பதால் மிகவும் சிக்கனமான முறையைக் கொண்டு கணிப்புகளைச் செய்தல் மிகவும் அவசியம். அதனால் கணிப்பு ஒழுங்கும் (order) கணிப்பிற்குப் பயன் படுத்தப்பட வேண்டிய பொறி களைப் பற்றிய முழு அறிவும் கொண்ட ஒருவரது பணி தேவைப் படுகிறது. இவரே சிஸ்டம்ஸ் அனலிஸ்ட் (Systems Analyst) எனப்படும் ஒருங்காய்வாளர் ஆவார்.

அடுத்தபடியாக மேல் நிலையில் உள்ளவர்களது பணி பொதுவாக நிலையத்தின் பணிகளைக் கட்டுப்படுத்தி ஒழுங்காகச் செயல்படச் செய்வதும், நிர்வாக சம்பந்தப்பட்ட பணிகளைக் கவனிப்பதும் ஆகும்.

3 கம்ப்யூட்டரும் வேலையின்மையும்.

கம்ப்யூட்டரைப் பற்றி யார் எந்த இடத்தில் பேசினாலும் அங்கே கேட்கப்படும் கேள்விகளுள், கம்ப்யூட்டரால் வேலை யின்மை ஏற்படாதா? அதுவும் இந்தியா போன்ற மக்கள் தொகை நிறைந்த நாட்டில் கம்ப்யூட்டரால் நன்மைகளைவிட, தீமைகள் தாம் அதிகம் விளையும் இல்லையா? என்பது ஒன்றாக உள்ளது. இதற்கு 'ஆம்' என்றோ 'இல்லை' என்றோ ஒரே வார்த்தையில் பதில் கூற இயலாது. இத்தகைய அச்சம் மனித இனத்திற்கு ஒன்றும் புதியது அல்ல. முதன் முதலாக புகை வண்டிகள் வந்த போது அவற்றிற்கு எதிர்ப்புக்கள் இருக்கத் தான் செய்தன. இந்த எதிர்ப்புக்கு மற்ற பலவேறு காரணங் களுடன், அங்கிருந்த குதிரை வண்டிக்காரர்களின் பிழைப்புக்

கெட்டுவிடும் என்ற அச்சமும் ஒன்ருக இருந்தது. பிரெஞ்சுப் புரட்சியின் போது, அன்றைய பிரெஞ்சு பார்லிமெண்டில் அங்கத்தினர் ஒருவர் மழுங்கிய கோடரிகளைச் சாணைபிடித்துத் திரும்பவும் பயன் படுத்தும் பழக்கத்தைத் தடைசெய்ய வேண்டும் என்று ஓர் திர்மானம் கொண்டுவந்தார். அவரது வாதம் கோடரிகளை சாணைபிடித்து திரும்ப உபயோகிப்பதால், புதிய கோடரிகளுக்கு தேவை குறைந்து, அவற்றை உற்பத்தி செய்வோர் பிழைப்பிழந்து விடுவார்கள் என்பதாம்.

இத்தகைய எதிர்ப்புகள் இருப்பினும் புதுவகையான பயன் தரத்தக்க கண்டுபிடிப்புகள் ஏதும் இவ்வெதிர்ப்புகளின் காரணமாக அழிந்து விடவில்லை. பார்க்கப் போனுல் அவைகள் பெருமளவுக்கு வளர்ந்து பயன்பட்டன. அதனுல் பாதிக்கப்பட்ட மக்கள்பகுதியினரும், சிறிதுசிறிதாக தங்கள் வாழ்க்கை, தொழில் முறைகளை மாற்றி அமைத்துக் கொண்டனர். மனித இனம் இத்தகைய கண்டு பிடிப்புகளின் விளைவுகளைத் தனது பரிணுமத்தின் (Evolution) ஒரு பகுதியாக ஏற்றுக்கொண்டு தன்னை சரிப்படுத்திக் கொண்டது. இத்தகைய பரிணும நிலைகள் (Stage of Evolution) யாவும் சமூகத்திற்கு வலியின்றி ஏற்பட்ட மாறுதல்கள் என்றும் கூறமுடியாது. பல்லாயிரக் கணக்கான மக்கள் தாம் பழக்கப்பட்டு வந்த தலைமுறை தலைமுறையாகச் செய்து வந்த தொழில்களை மாற்றி வேறு தொழில்களை மேற்கொண்டு பிழைப்பு நடத்துவதென்பது எளிதல்ல. அப்படியே அவர்கள் தயாராக இருப்பினும், அவர்களுக்கு வேண்டிய மாற்றுத் தொழில்கள் கிடைக்க வகை வேண்டுமே. ஒரு காலத்தில் இத்தகைய சமூக வலியை யாரும் பொருட்படுத்துவதில்லை; அது இயற்கையாகவே குணமடைய வேண்டியிருந்தது. பாதிக்கப்பட்ட மக்கள் அழிந்தாலோ அல்லது, தாமாகவே வேறு தொழில் செய்து வாழ்ந்தாலோ, புதிய கண்டு பிடிப்புகளால் பாதிக்கப்பட்ட தொழில்கள் சிறிது சிறிதாக நசித்தன. ஆனுல் இன்றைய அரசுகளும், சமூக அமைப்புகளும், குறிப்பாக 'மக்கள் நல' அரசுகளும் (Welfare States), இத்தகைய சமூக பரிணுமத்தை வலியற்றதாகச் செய்யக் கடமைப்பட்டுள்ளன.

கம்ப்யூட்டர்களால் மனிதர்களின் தொழில் வாய்ப்புக்கள் பாதிக்கப்படுமோ என்ற அச்சம் இங்கு மட்டும் உள்ளதல்ல. வேலை வாய்ப்புக்கள் மிகுந்து மக்கள் தொகை குறைந்துள்ள அமெரிக்க ஐக்கிய நாட்டின் மக்களிடையேயும் கூட இந்த அச்சம் இல்லாமல் போகவில்லை. இத்தகைய அச்சத்திற்கு காரணம், கம்ப்யூட்டரைக் கண்டு அஞ்சுவோர் அதைப் பற்றிய

உண்மையான அறிவு பெற்றிராததும், அதேசமயத்தில் கம்ப் யூட்டரைப் பயன்படுத்துவோர் அதை வரைமுறையின்றிப் பயன் படுத்த முயலும் தன்மையும் ஆகும்.

கம்ப்யூட்டரைப் பயன்படுத்திச் செய்யப்படும் கணிப்புகளை இரண்டு வகையாக பிரிக்கலாம். ஒன்று மனிதர்களால் உரிய காலத்தில் செய்து முடிக்க இயலாதது. இரண்டாவது, மனிதர் களால் குறித்த காலத்தில் செய்து முடிக்கக் கூடியது. இவற் றுள் முதல் வகையைச் சேர்ந்த பணிகளில் கம்ப்யூட்டர்களைப் பயன்படுத்துவதுபற்றி எவரும் எந்தக் குறையும் கூற இயலாது; இத்தகைய கணிப்புகளில் கணிப் பொறிகளைப் பயன்படுத்து வதால் எவருக்கும் வேலை போவதில்லை. எடுத்துக் காட்டாக சிலவற்றைக் கூறலாம். விஞ்ஞான ஆராய்ச்சித் துறைகள் யாவற்றிலும் கம்ப்யூட்டர்களைத் தங்குதடையின்றிப் பயன்படுத்த லாம். இன்றுள்ள விஞ்ஞானத் துறைகள் பலவும் தனது இன் றைய வளர்ச்சி வேகத்திற்கு, கம்ப்யூட்டருக்குப் பெரிதும் கடன் பட்டுள்ளன. இவ்வாறே பாதுகாப்பு, வானிலே முன் கணிப்பு ஆகிய துறைகளையும் கூறலாம். இத்துறைகளில் உள்ள கணிப்புகள் யாவும் பெரும்பாலும் மனிதர்களின் கைக்கணிப்பிற் கப்பாற்பட்டவை. இத்துறைகளில் கம்ப்யூட்டர்களைப் பயன் படுத்துவதால் நன்மை மட்டுமே விளையும்.

தொழில், வணிகத்துறைகளை எடுத்துக் கொண்டால் அங்கு முன் கூறிய இரண்டு வகையான கணிப்புகளையும் காணலாம். சம்பளப் பட்டியல், விற்பனைப் பட்டியல் போன்ற கணிப்புகள் போன்றவை நாள்தோறும், அல்லது வாரம், மாதந்தோறும் செய்யப்படும் சாதாரண கணிப்பு ஆகும். இத்தகைய கணிப்பு களை மனிதர்களின் உழைப்பைக் கொண்டே உரிய காலத்திற் குள் கணித்துக் கொள்ளவும் முடியும். இத்தகைய கணிப்பு களுக்கு கம்ப்யூட்டர்களை மக்கள் தொகை பற்ருத, வேலைப் படை (work force) குறைந்த நாடுகளில் பயன்படுத்தலாம். ஆனல் வேலையின்மை நிறைந்த வளரும் நாடுகளில் இவற்றிற்கு கம்ப்யூட்டர்களைப் பயன்படுத்துவது தவறேயாகும். எந்தெந்த கணிப்புகளுக்குக் கம்ப்யூட்டரைப் பயன்படுத்தலாம், எங்கெங்கு பயன்படுத்தலாம் என்பதோ அறுதியிட்டுக் கூறமுடியாதவையா கும். சுருங்கக்கூறின் எங்கெங்கு மனித உழைப்பைக்கொண்டு, கணிப்புகளைத் திறம்பட செய்ய இயலாதோ அங்கு மட்டும் கம்ப்யூட்டர்களைப் பயன்படுத்துவது சாலச் சிறந்ததாம்.

கம்ப்யூட்டர்களை எல்லா துறையிலுமே பயன்படுத்த வேண் டும் என்று கூறுவோர் மற்ருெரு வகையிலும் விவாதிக்கிருர்

கள். அவர்களது விவாதத்தை கீழ்வருமாறு சுருக்கமாகக் கூற லாம். புகைவண்டிகளும், பஸ்களும், போக்குவரத்தின் பெரும் பகுதியை ஏற்றுக் கொண்டபோது, ஜட்கா வண்டிக்காரர்கள் போன்ற மெதுவான போக்கு வரத்துச் சாதனங்களை வைத்துப் பிழைப்பு நடத்தியவர்கள் வேலையிழந்தார்கள் என்பது உண்மை தான். ஆனால் புகைவண்டிகள் வந்ததால் அதுவரை இல்லாத பல புதிய தொழில்கள் ஏற்பட்டன. இப்புதிய தொழில்களில் வேலை வாய்ப்புப் பெற்றோரின் எண்ணிக்கை, வேலை இழந் தோரின் எண்ணிக்கையை விட மிகவும் அதிகம்; இம்முறையில் பார்க்கப் போனால் மொத்தத்தில் அப்புதிய கண்டுபிடிப்பால் அதிகப்படியான வேலைவாய்ப்புக்கள் ஏற்பட்டனவே அன்றி வேலையிழப்பு ஏற்படவில்லை. அதுவே போல கம்ப்யூட்டர்கள் வந்த பிறகு கைக்கணிப்பு சம்பந்தப்பட்ட வேலைகளில் ஈடுபட் டிருப்போர் வேலையிழக்க நேரிடலாம்; ஆனால் அதே சமயத்தில் கம்ப்யூட்டர் வந்த பிறகு தொழில் துறைகள் யாவும் அளவில் பெருகியதில் மட்டுமல்லாமல், பலவேறு புதிய தொழில் துறை களும் தோன்றி முன்னிருந்ததை விட அதிகப்படியான வேலை வாய்ப்புக்கள் ஏற்படும். அதனால் கம்ப்யூட்டரை 'மனிதர்களின் வேலை போகிறது' என்ற அச்சம் ஏதுமின்றி தாராளமாக பயன் படுத்தலாம். இத்தகைய விவரத்தில் உண்மையில்லாமல் போகவில்லை. ஆனால் ஒன்றைக் கருத்தில் கொள்ள வேண்டும். வேலை இழப்போரும், புதிய வேலை வாய்ப்பு பெறுவோரும் இருவேறு பிரிவினர் என்பதை நாம் உணரவேண்டும். கூட்டிக் கழித்துப் பார்க்கும் போது அதிகப்படியான வேலை வாய்ப்பு இருப்பது உண்மையானதாக இருக்கலாம். ஆனால் இந்தக் கூட்டிக் கழித்தலில் ஒருபுறத்தில் வேலையிழப்போரே மறுபுறத் தில் புதிய வேலை வாய்ப்புபெற வசதியிருக்க வேண்டும். இது மட்டும் போதாது; ஒரு நாட்டில் ஆண்டு தோறும் புதிதாக வேலை பெறுவோரின் விகிதாசாரமும் குறையாமல் இருத்தல் அவசியம். எடுத்துக் காட்டாக ஒரு நாட்டில் ஆண்டுதோறும் சராசரி புதிதாக 1000 பேர் புதிதாக வேலை பெறுவார்கள் என்று வைத்துக் கொள்ளுவோம். அந்நாட்டில் ஒரு குறிப்பிட்ட நாளில் அவர்களது பெரும்பாலான அலுவல்களை கம்ப்யூட்டர் மயமாக்குவதாக வைத்துக் கொள்ளுவோம். அவ்வாறு செய்யும் போது, பாதிக்கப்பட்ட அலுவலர்கள் யாவரும் புதிய பயிற்சி அளிக்கப்பட்டு கம்ப்யூட்டர் நிலையங்களிலோ, அல்லது புதி தாகத் தோன்றிய தொழில் துறைகளிலோ எடுத்துக் கொள்ளப் படுகிறார்கள் என்றும் கொள்ளுவோம். அதாவது ஏற்கனவே வேலையிழந்தோர் எவரும் தமது வேலையிழக்கவில்லை. ஆனால்

அதே சமயத்தில் அடுத்த ஆண்டு புதிதாக வேலை பெறுவோரின் எண்ணிக்கை சராசரியான 1000-லிருந்து கணிசமான அளவிற்குக் குறைந்தால் - இவ்வேலை வாய்ப்புக் குறைவு கம்ப்யூட்டர்களால் ஏற்பட்டிருந்தால் அது விரும்பத்தக்கதல்ல. ஆக சுருக்கமாகக் கூறினல், கம்ப்யூட்டரைப் பயன்படுத்துவதால் வேலையிழப்போ அல்லது வேலை வாய்ப்பில் குறைவோ ஏற்படாமலிருப்பின் மட்டுமே அவற்றைக் குறிப்பிட்ட துறைகளில் பயன்படுத்துவதை அனுமதிக்கலாம். இந்தியா போன்ற, மக்கள் தொகையும் அதன் சார்பான வேலையின்மை போன்ற பிரச்சினைகளும் மலிந்த நாடுகளில் அரசாங்கமே கம்ப்யூட்டர் பயன்படுத்தப்பட வேண்டிய துறைகளைக் கண்காணித்து, உரிய காலம் வரை, உரிய கட்டுப்பாடுகளை ஏற்படுத்தி, கம்ப்யூட்டர்களின் நுழைவால் ஏற்படும் சமூக வேலை வாய்ப்பும் பரிணாமத்தை 'வலியற்ற'தாகச் செய்யக் கடமைப்பட்டுள்ளன.

4 கம்ப்யூட்டர் ஒருங்கு வகைகள் (Types of Computer Systems)

இன்றுள்ள கம்ப்யூட்டர் நிலையங்கள் பலவகையான ஒருங்குகளைக் கொண்டுள்ளன. அவற்றை அணிமுறை ஆய்வு ஒருங்கு (Batch Processing System) பல் ஆய்வங்க ஒருங்கு (Multi Processing System) பல் புரோகிராம் ஒருங்கு (Multi Programming Systems) நிகழ்நேர ஒருங்கு (Real Time Systems) என நான்கு முக்கிய வகைகளாகப் பிரிக்கலாம்.

அணிமுறை ஆய்வு (Batch Processing)

சாதாரணமாக ஒரு கம்ப்யூட்டரில் ஒன்றன் பின் ஒன்ருக பல புரோகிராம்களை ஓட்ட வேண்டி இருக்கும். ஒரு புரோகிராம் ஆய்வதற்கு, முதலில் அந்தப் புரோகிராமிற்கான கம்ப்பைலர் அல்லது புராஸஸர், கம்ப்யூட்டரின் நினைவகத்தில் ஏற்றப்பட வேண்டும். அதன் பின்னர் நமது புரோகிராம் கார்டுகள் உள்ளே அனுப்பப்பட்டு மொழி பெயர்ப்பு, கணிப்பு, முடிவாக விடையிறுப்பு ஆகியவை நடை பெறுகின்றன. இவ்வாறு ஒரு புரோகிராம் முடிந்தவுடன், மற்றொரு புரோகிராமை ஆய மீண்டும் மேற்கண்ட முறைகளைத் திரும்பிக் கையாள வேண்டும். இவ்வாறு புரோகிராம் ஆய்வதில் ஒரு பெரிய அசௌகரியம் கம்ப்யூட்டரின் காலவிரயமாகும். ஒவ்வொரு கார்டுகளை, கையால் எடுத்துப் போடும் போதும் குறிக்கத்தக்க அளவு கால விரயம் ஏற்படுகிறது. கணிக்கப்படப் போகும் புரோகிராம்கள் சிறியவையாக இருப்பின், கணிப்பதற்காக ஆகும் நேரம், அந்தப்புரோகிராமைக் கையிலெடுத்து படிப்பொறியிலிட்டு ஓட்டு

வதற்கு ஆயத்தம் செய்யும் நேரத்தைவிடக் குறைவாக இருக்கக்கூடும்; அதுவுமல்லாமல் இம்முறையில் கம்ப்யூட்டர் ஓட்டுவோரின் இடையருத கவனமும் தேவைப்படுகிறது.

மேற்கண்ட முறையில் ஆகும் காலவிரயத்தைத் தடுக்க 'அணிமுறை ஆய்வுமுறை' ஏற்பட்டது. இம்முறைப்படி கணிப்பிற்கான புரோகிராம்கள் முதலில் ஒரு அணியாக காந்த நாடாவில் (அல்லது தட்டில்) துளைக் கார்டுகளிலிருந்து எழுதிக் கொள்ளப் படுகின்றன. அதன்பின்னர் காந்த நாடாச் சுருளிலிருந்து புரோகிராம்கள் கம்ப்யூட்டருக்கு ஒவ்வொன்றாக ஊட்டப் படுகின்றன. முதல் புரோகிராம் கம்ப்யூட்டருக்குள் சென்றதும், மொழிபெயர்த்து சீரமைக்கப்பட்டு (Compiled) கணிக்கப்படுகின்றது. அப்போது தவறுகள் ஏதும் காணப்படின் அந்தப் புரோகிராமை விட்டுவிட்டு அடுத்த புரோகிராமை தானாகவே காந்த நாடாவிலுருந்து எடுத்துக் கொண்டு மீண்டும் செயல்பட ஆரம்பிக்கிறது. ஆனால் அதற்கு முன்னர் தவறு ஏற்பட்ட புரோகிராமையும் தவறு பற்றிய விவரங்களையும் கம்ப்யூட்டர் வேறு ஒரு புதிய காந்த நாடாவில் எழுதிக் கொள்ளுகிறது. இரண்டாவது புரோகிராமில் தவறு ஏதுமில்லை என்று வைத்துக் கொள்ளுவோம். அந்தப் புரோகிராம் முற்றும் கணிக்கப்பெற்று அதன் இறுதி விடைகள் கிடைத்தவுடன், புரோகிராமையும் அவ்விடைகளையும் கம்ப்யூட்டர் முன்கூறப்பட்ட கொடுதுறை (Out put) காந்த நாடாவில் எழுதிக் கொள்ளுகிறது. இவ்வாறு எழுதப்பட்ட புரோகிராமும் அதன் விடைகளும், காந்தநாடாவில் முதல் புரோகிராமும் அதில் காணப்பட்ட தவறுகளும் எழுதப்பட்டுள்ள இடத்தை அடுத்து எழுதப்பெறுகின்றன. இவ்வாறாக கொள் நாடா (Input Tape) வில் உள்ள எல்லா புரோகிராம்களும் ஒன்றன் பின் ஒன்றாக எடுத்துக் கொள்ளப்பட்டு, அவற்றிற்குரிய விடைகளோ அல்லது பிழைக் குறிப்புகளோ கொடு நாடா (Out put Tape) வில் எழுதப்படுகின்றன. இத்தகைய ஆய்வு முற்றுப் பெற்றவுடன் கொடு நாடாவில் உள்ள செய்திகள் யாவும் ஒரே மூச்சில் அச்சுப்பொறியில் அச்சடிக்கப்பட்டு புரோகிராம்களுக்கு உரியவர்களிடம் சேர்க்கப்படுகின்றன.

மேற் சொன்ன முறையைக் கையாளும் நிலையங்களில் பெரும்பாலும் குறைந்தது இரண்டு கம்ப்யூட்டர்கள் இருக்கும். இவற்றுள் ஒன்றின் கணிப்பு வேகம் மிக அதிகமாகவும், மற்றதின் வேகம் குறைந்ததாகவும் இருக்கும். இவற்றை முறையே பெரிய, சிறிய கம்ப்யூட்டர்கள் என்று குறிப்பிடுவோம். சிறிய கம்ப்யூட்டரின் பணி, கார்டுகளில் உள்ள புரோகிராம்களை காந்த

நாடாவில் ஏற்றி புரோகிராம் அணிகளைத் (Batches) தயாரிப்ப தாம். இந்த காந்த நாடாக்கள் பெரிய கம்ப்யூட்டரில் இடப் பட்டு (கொள்துறையாக - Input) - முன் சொன்னவாறு புரோக் கிராம்களின் விடைகள் அல்லது பிழைக்குறிப்புக்கள் கொண்ட கொடுநாடாக்கள் தயாரிக்கப்படுகின்றன. இக்கொடு நாடாக்கள் மீண்டும், சிறிய கம்ப்யூட்டருக்கு எடுத்து செல்லப் பட்டு, அதன் உதவியால் இந்நாடாவில் காணும் விவரங்கள் சிறிய கம்ப்யூட் டருடன் இணைக்கப்பட்டுள்ள அச்சுப் பொறியில் அச்சிடப்படு கின்றன. இவ்வாறு அச்சிடப்பட்ட விவரங்கள் இறுதியாக புரோகிராமிற்கு உரியவர்களிடம் சேர்க்கப்படுகின்றன.

மேற்கண்ட முறையில் நாம் நோக்கத்தக்கவை சில உள்ளன. ஒன்று பெரிய கம்ப்யூட்டரில் காலம் விரயமாக்கப் படுவதில்லை. மற்றொன்று காந்த நாடாவில் ஏற்றப்பட்டு பெரிய கம்ப்யூட்டரால் ஆயப்படும். புரோகிராம்கள் ஒரு கியூ வரிசை யில் நிறுத்தப்பட்டு ஒவ்வொன்றாக ஆயப்படுகின்றன; ஒரு புரோ கிராமில் தவறுகள் இருந்தாலோ அல்லது அது வெற்றிகரமாக ஆய்ந்து முடிக்கப்பட்டாலோ அன்றி அடுத்தப் புரோகிராம் எடுத்துக் கொள்ளப்படமாட்டாது. இந்த முறையில் ஆயப்படும் கம்ப்யூட்டர் நிலையத்தில் நாம் ஒரு புரோகிராமைக் கொடுத்தால் நமக்கு அந்தப் புரோகிராம் பிழைக்குறிப்புகளுடனோ அல்லது விடைகளுடனோ நமக்குத் திரும்பிவர ஆகும் நேரம் -(இதை Turn Around Time என்பர். நாம் இதைத் 'திருப்பு நேரம்' என்போம்.) எவ்வளவு என்று அறுதியிட்டுக் கூற முடியாது. வரிசையில் நமது புரோகிராமிற்கு முன்னராக எத்தனை புரோ கிராம்கள் உள்ளன என்பதையும் ஒவ்வொரு புரோகிராமிற்கும் ஆகும் கணிப்பு நேரத்தையும் பொறுத்தே நமது புரோகிராமின் திருப்பு-நேரம் அமையும். இது சில நிமிடங்களாகவும் இருக்க லாம்; சில நாட்களாகவும் ஆகலாம்.

பல் புரோகிராம் கம்ப்யூட்டர் ஒருங்கு.

எந்த ஒரு கம்ப்யூட்டர் ஒருங்கிலும், மத்திய ஆய்வங்கமும், கொள்/கொடு அங்கங்களும் இன்றியமையாத பாகங்களாம். இவற்றுள் மத்திய ஆய்வங்கத்தின் கணிப்பு வேகம் மிக அதிக மாகும். இன்றைய கம்ப்யூட்டர்களின் கணிப்பு வேகங்கள் நானோ செகண்டுகளில் (ஒரு நானோ செகண்டு என்பது $\frac{1}{10^9}$ செகண்டு ஆகும்.) கணக்கிடப்படுகின்றன. இந்த வேகத்தை ஒப்பு நோக்க கொள்/கொடு அங்கங்களின் செயல் வேகம் மிக மிகக் குறைவாகக் காணப்படும். சில கணிப்புகளை இன்றைய கம்ப்யூட்டர் $\frac{1}{2}$ விநாடியில் முடித்து விட முடியும். ஆனால் இந்த

கணிப்பின் முடிவுகளை (Results) ஒரு இணையச்சுப் பொறி (On time Printer) வாயிலாகவோ அல்லது கார்டு துளைப் பொறிகள் வாயிலாகவோ வெளியிட 5 அல்லது 6 நிமிடங்கள் வரை கூட ஆகலாம். சாதாரணக் கம்ப்யூட்டர், அமைப்பில், மத்திய ஆய் வங்கம், கொள்/கொடு அங்கங்கள் இவையாவும் ஆளங்கம் ஒன்றின் ஆளுகைக்கு உட்பட்டு இயங்கும். அந்நிலையில் மத்திய ஆய்வங்கத்தில் ஆயப்பட்ட இறுதி அல்லது இடைநிலை (Intermediate) முடிவுகள் அச்சுப் பொறியாலோ அல்லது துளைக் கார்டுகள் மூலமாகவோ வெளியிடப்பட்டுக் கொண்டிருக்கும் போது (அதாவது கொள்/கொடு துறைகள் இயங்கிக் கொண்டு இருக்கும் நிலையில்) மத்திய ஆய்வங்கம் செயல்படாமல் இருக்கும். இக்காரணத்தால் மத்திய ஆய்வங்கத்திற்கு ½ விநாடி வேலை 5 நிமிட ஓய்வு, பின்னர் ¼ விநாடி வேலை, 4 நிமிட ஓய்வு என்பன போன்ற நிலை ஏற்பட இடமேற்படுகிறது. மத்திய ஆய்வங்கம் விலை மிகுந்த ஒரு பாகமாகும். இதை செயலற்று சோம்பி வைத்திருக்கும் ஒவ்வொரு விநாடியும், கால விரயமும், பொருள் நட்டமும் ஏற்படுகிறது. இத்தகைய நிலையை தவிர்த்து மத்திய ஆய்வங்கத்தை மேலும் பயனுள்ள வகையில் உபயோகப்படுத்தும் முகத்தான் பல்புரோகிராம் கம்ப்யூட்டர்கள் வந்துள்ளன. இக்கம்ப்யூட்டர்கள் எவ்வாறு செயல்படுகின்றன என்பதைச் சுருக்கமாக பார்ப்போம்.

பல்புரோகிராம் கம்ப்யூட்டர்களின் நினைவகம் பல பிரிவுகளாகப் பிரிக்கப்பட்டிருக்கும். இப்பிரிவு ஒவ்வொன்றும் ஒரு தனி நினைவகத்தைப் போல் இயங்க வல்லது. எடுத்துக் காட்டாக இத்தகைய ஒரு நினைவகம் ஐந்து பாகங்களாகப் பிரிக்கப்பட்டிருக்கின்றது என்று வைத்துக்கொள்ளுவோம். (இதை Paging என்று கூறுவர்) இந்த ஐந்து பாகங்களும் ஐந்து சிறிய தனி நினைவகங்களைப் போல இயங்கவல்லன. ஒரே சமயத்தில் நாம் ஐந்து புரோகிராம்களை இக்கம்ப்யூட்டருக்கு ஊட்ட முடியும். இவ்வைந்து புரோகிராம்களும் நினைவகத்தின் ஐந்து பகுதிகளில் தனித்தனியே போய் உட்காருகின்றன. இப்புரோகிராம்களைக் கம்ப்யூட்டர் ஆயும் நிலையில், முதன் முதலாக நினைவகத்தின் முதல் பாகத்தில் வந்து சேர்ந்துள்ள புரோகிராமை எடுத்துக் கொண்டு அதைக் கணிக்கிறது. இப்புரோகிராமின் கணிப்பினூடே கணிப்பின் இடை முடிவுகளை (Intermediate-Results) அச்சிட்ட பின்னர் மீண்டும் கணிப்பைத் தொடருமாறு புரோகிராம் அமைந்திருப்பதாகக் கொள்ளுவோம். இடை முடிவுகளை அச்சிடுவதற்கான ஆணை எதிர்ப்பட்டதும் கம்ப்யூட்டர் இம்முடிவுகளை கொடு அங்கத்திற்கு (அச்சுப் பொறி அல்லது

துளைப்பொறி) அனுப்பி விடுகிறது. கொடு அங்கம் இவ்விடைகளை ஒட்டு மொத்தமாகப் பெற்றுக் கொண்டு அவற்றை அச்சிட அல்லது கார்டுகளில் துளைக்க ஆரம்பிக்கிறது. இப்போது மத்திய ஆய்வங்கம், மேற்சொன்ன விடைகள் அச்சிடப்படும் வரைக் (அல்லது துளைக்கப்படும் வரை) காத்திருப்பதில்லை. அச்சிட வேண்டியவற்றைக் 'கொடு' அங்கத்தில் கொண்டு சேர்த்த உடனே, அது நினைவகத்தின் இரண்டாம் பகுதியிலுள்ள புரோகிராமை எடுத்துக் கொண்டு கணிக்க ஆரம்பிக்கிறது. இந்நிலையில் முதல் புரோகிராமின் இடைநிலை விடைகள் 'கொடு' அங்கப்பொறியான அச்சுப் பொறியில் அச்சாகிக் கொண்டிருக்கும் அதே நேரத்தில், மத்திய ஆய்வங்கம் இரண்டாம் புரோகிராமைக் கணித்துக் கொண்டிருக்கும் இதே முறையில் இரண்டாம் புரோகிராமில் 'கொடு' ஆணை ஒன்று (Output Instruction) எதிர்ப்பட்டதும், அச்சிட வேண்டியவற்றை அச்சுப் பொறிக்கு மொத்தமாக அளித்து விட்டு, நினைவகத்தின் மூன்றும் பகுதியில் உள்ள புரோகிராமை எடுத்துக் கொண்டு செயல்படுகிறது. பல் புரோகிராம் கம்ப்யூட்டர்களில் ஒன்றுக்கு மேற்பட்ட 'கொடு' அங்கங்கள் சாதாரணமாகக் காணப்படும். ஒரு கம்ப்யூட்டருடன் இரண்டு மூன்று அச்சுப் பொறிகளும், ஒன்றுக்கு மேற்பட்ட துளைப்பொறிகளும் இணைக்கப்பட்டிருக்கும். ஒரு புரோகிராமின் இடை, இறுதி விடைகள் ஒரு அச்சுப் பொறியில் அச்சாகிக் கொண்டிருக்கும் போதே மற்றொரு புரோகிராமிற்கான விடைகளும் அச்சிடப்பட வேண்டிய தேவை ஏற்படலாம். இந்நிலையில் கம்ப்யூட்டருடன் இணைக்கப்பட்டுள்ள மற்றொரு அச்சுப் பொறி இந்த இரண்டாவது புரோகிராமின் விடைகளை அச்சிடும் பணியை மேற்கொள்ளுகிறது.

மேற்கண்ட முறையில் கம்ப்யூட்டர் நினைவகத்தில் உள்ள ஐந்து புரோகிராம்களையும் 'கவனித்து' முடிந்த பிறகு மீண்டும் முதல் புரோகிராமிற்கு வந்து அதன் கணிப்பை விட்ட இடத்திலிருந்து தொடருகிறது. மீண்டும் இப்புரோகிராமில் 'கொடு' ஆணை எதிர்ப்பட்டதும், முன் கூறியவாறு அச்சிடவேண்டியவற்றை அச்சுப் பொறிக்கு அளித்து விட்டு இரண்டாவது புரோகிராமிற்கு செல்லும்.

மேற்கூறிய வகைக் கம்ப்யூட்டர்களில் ஒன்றுக்கொன்று சிறு வேறுபாடுகளைக் கொண்டிருக்கும். சிலவற்றில் முன் சொன்ன நினைவகத்தின் பகுதிகள் நிலையாகப் பிரிக்கப்பட்டிருக்கும். அதனால் ஒவ்வொரு பகுதியின் கொள்ளளவும் (Capacity) நிலையான ஒன்றாகும். இத்தகைய கம்ப்யூட்டர்களில் நாம் கணிக்க விரும்

பும் எந்தப் புரோகிராமிற்கும் அதற்கு தேவையான நினைவறைகளின் எண்ணிக்கை நினைவகத்தின் பகுதிகளின் தனித்தனிக் கொள்ளளவுக்கு (Individual Capacity) உட்பட்டதாயிருத்தல் அவசியமாகும். எடுத்துக்காட்டாக, கம்ப்யூட்டரின் நினைவகப் பகுதி ஒவ்வொன்றும் 40,000 நினைவறைகளைக் கொண்டது என்று வைத்துக் கொண்டால், நமது புரோகிராம் ஒவ்வொன்றிற்கும் தேவையான நினைவறைகளின் எண்ணிக்கை 40,000க்கும் மேற்படக் கூடாது.

மற்றும் சில பல்புரோகிராம் கம்ப்யூட்டர்களில் நினைவகத்தின் பகுதிகள் நிலையாகப் பிரிக்கப்பட்டிருப்பதில்லை. கம்ப்யூட்டருக்கு ஊட்டப்படும் புரோகிராம்களுக்குத் தேவைப்படும் நினைவறைகளின் எண்ணிக்கைக்கு தக்கவாறு அவ்வப்போது நினைவகம் பகுதிகளாகப் பிரிக்கப்படும். எடுத்துக் காட்டாக நாம் மூன்று புரோகிராம்களை ஆய வேண்டும் என்று வைத்துக்கொள்ளுவோம். இவற்றிற்கு முறையே 25,000, 32,000 57,000 நினைவறைகள் தேவைப்படுகின்றன என்று கொண்டால், கம்ப்யூட்டர் தானாகவே 25,000, 32,000, 57,000 நினைவறைகளைக் கொண்ட மூன்று பகுதிகளாகப் பிரித்துக் கொண்டு செயல்படுகின்றது. இந்த முறை முன் முறையைக் காட்டிலும் சிறந்தது என்பது கூறாமலே விளங்கும்.

மேற்சொன்ன கம்ப்யூட்டர்களில் வேறு ஒரு வசதியும் உண்டு; நினைவகத்தின் பகுதிகளில் உள்ள பல புரோகிராம்களும், ஒவ்வொன்றாக வரிசை முறையில் ஆயப்பட்டு, 'கொடு' ஆணை ஒன்று எதிர்பட்டதும், அந்தப் புரோகிராமை விட்டு அடுத்த புரோகிராமிற்குப் போகிறது என்று பார்த்தோம். ஆனால் சில வகைக் கம்ப்யூட்டர்களில் ஆயப்படும் புரோகிராம் ஒவ்வொன்றிற்கும் குறிப்பிட்ட கால அளவு பங்கீடு செய்யப்பட்டு விடும். எடுத்துக் காட்டாக 5 புரோகிராம்கள், நினைவகத்தில் இருப்பின் ஒவ்வொன்றிற்கும் 0.25 விநாடி என்று கால அளவு பங்கீடு செய்யப்பட்டிருப்பின், முதல் புரோகிராமிற்கான கணிப்புகளை 0.25 விநாடி செய்யும்; இந்த கால அளவு முடிந்தவுடன் அந்தப்புரோகிராமின் நிலை எதுவாயிருப்பினும் அதை அப்படியே விட்டு விட்டு, அடுத்தப் புரோகிராமை எடுத்துக் கொள்ளும். இந்த இரண்டாவது புரோகிராமையும் 0.25 விநாடி காலம் கணித்த பிறகு அதை நிறுத்தி விட்டு அடுத்த புரோகிராமிற்கு போகிறது. இவ்வாறு 5 புரோகிராம்களையும் ஒவ்வொன்றிற்கும் 0.25 என்ற காலக் கணக்கில் கணித்து முடித்த பின்னர் மீண்டும் முதல் புரோகிராமிற்கு வந்து அதை விட்ட இடத்திலிருந்து

தொடர்ந்து 0.25 விநாடி கணிக்கிறது. மீண்டும் இக்காலக் கெடு முடிந்ததும் முதல் புரோகிராமை விட்டு விட்டு அடுத்தப் புரோகிராமிற்குச் செல்லும். இத்தகைய கணிப்பினூடே ஒரு புரோகிராமிற்குத் தேவையான மொத்த நேரமே 0.25 விநாடிக் கும் குறைவானதாயிருப்பின்—எடுத்துக் காட்டாக 0.15 விநாடி என்றால் இந்தப் புரோகிராம் எடுத்துக் கொள்ளப்பட்டு அதன் ஆய்வு முடிந்த உடனே கம்ப்யூட்டர் அடுத்த புரோகிராமிற்குப் போய்விடும். அதாவது ஒவ்வொரு முறையும் ஒரு புரோகிராமும் 0.25 விநாடி அல்லது அதை முடிப்பதற்கு ஆகும் நேரம் இவற்றுள் எது குறைவோ அந்த அந்த அளவு நேரமே கணிக் கப்படுகிறது.

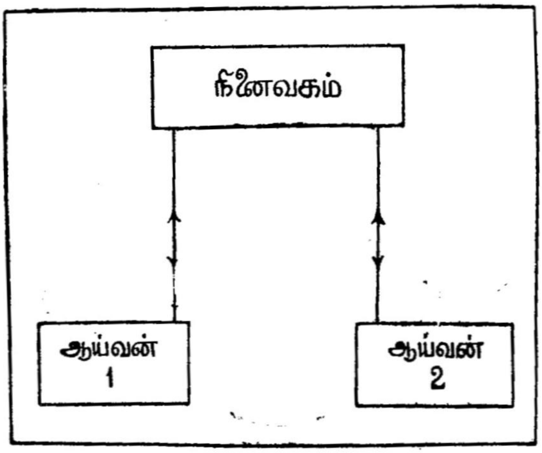
சில சமயங்களில் பல புரோகிராம்களை ஒன்றாக ஆயும் போது, ஒரு புரோகிராம் மிகவும் அவசரமானதாக இருக்கலாம். அந்நிலையில் அந்தப் புரோகிராமும் மற்றவற்றைப் போல கியூ முறையில் ஆயப்படுமாயின் கால தாமதம் ஏற்படக்கூடும். இதைத் தவிர்க, இப்புரோகிராமிற்கு முக்கியத்துவம் கொடுத்து அதற்கு அதிக நேரம் அளித்து அதை முதலில் முடித்துக் கொடுக்குமாறு கம்ப்யூட்டருக்கு ஆணையிடவும் இயலும். ஒரு புரோகிராமிற்கு மட்டுமல்லாமல், ஒரே சமயத்தில் ஆயப்படும் எல்லா புரோகிராம்களுக்கும் வெவ்வேறு அளவில் முக்கியத் துவம் இருக்கலாம். புரோகிராம்களின் முக்கியத்துவங்களுக்கு (Priorities) ஏற்றவாறு அவற்றிற்கு ஏற்ற நேரமும், புரோகிராம் கியூவில் அதிகப்படியான வாய்ப்பும் அளிக்க இயலும்.

மேற்கண்ட பல் புரோகிராம் கம்ப்யூட்டர் அமைப்புகளில் தொலை முனை (Remote Terminal) வசதிகளும் உண்டு. எடுத் துக் காட்டாக சென்னையில் இருக்கும் கம்ப்யூட்டர் ஒன்றுடன் செங்கல்பட்டு, அரக்கோணம், வேலூர் ஆகிய ஊர்களில் உள்ள அலுவலகங்களில் டெலிபிரிண்டர்களில் சில மாறுதல்களைச் செய்து இணைத்துக் கொள்ளலாம். இத்தகைய இணைப்புகளைத் தொலைமுனைகள் (Remote Terminals) என்பர். இத்தகைய தொலைமுனைகள் வழியாகக் கம்ப்யூட்டருக்கும் புரோகிராம்களை அளித்து விடைகளையும் பெற இயலும். இத்தகைய அமைப்பில் உள்ள மையக் கம்ப்யூட்டர், பல்புரோகிராம் வசதி உள்ளதாய் இருப்பதால், அது ஒரேசமயத்தில் பல வேறு தொலை முனைகளிலி ருந்தும் வரக்கூடிய புரோகிராம்களையும் ஏற்றுக் கணிக்கிறது. இத்தகைய அமைப்பில் ஒரு மனோதத்துவ ரீதியான சௌகரியம் உண்டு. மையக் கம்ப்யூட்டரைப் பயன்படுத்தும் நூற்றுக்கணக் கானவர்களில் தொலை முனை வைத்திருப்போர் ஒரு பகுதியாவர்,

ஆனால் தொலை முனையில் உட்கார்ந்துக் கொண்டு ஒரு புரோ கிராமை ஒருவர் கம்ப்யூட்டருக்கு அனுப்பிக் கணிப்பைச் செய் யும் போது, தனக்கு மட்டுமே உரிய ஒரு கம்ப்யூட்டரைப் பயன் படுத்திக் கொண்டிருப்பதுப் போன்ற ஒரு உணர்வையே அவர் பெறுகிறார்.

பல ஆய்வன் ஒருங்கு. (Mutli Processing System)

சாதாரணமாக ஒரு கம்ப்யூட்டரில் ஒரு நினைவகமும், ஒரு ஆய்வனும் (Processing Unit) காணப்படும். ஆனால் ஒரு குறிப் பிட்டவகை ஒருங்கில் ஒரு பெரிய நினைவகமும் அதோடு ஒன்றுக்கு மேற்பட்ட ஆய்வன்களும் இணைக்கப்பட்டிருக்கும். (ஆய்வன் என்பது ஒரு கணிப்பங்கமும் ஒரு ஆளங்கமும் கொண்டது.)

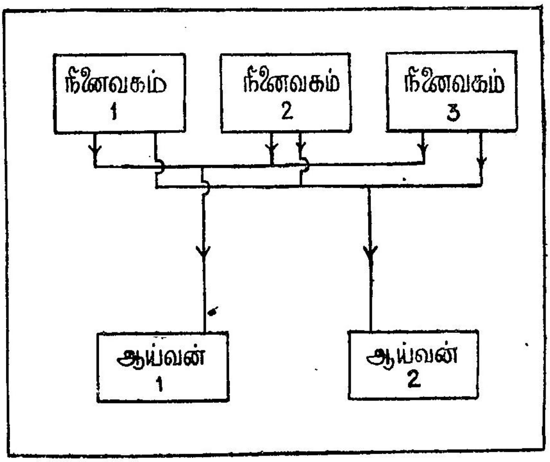


படம் 7.1

பல் — ஆய்வன் ஒருங்கு — முதல்வகை

இத்தகைய கம்ப்யூட்டர்களில் ஒரே சமயத்தில் ஒன்றுக்கு மேற்பட்ட கணிப்புகள் நடக்க வசதியுள்ளது. ஆய்வன் 1 நினை வகத்தில் உள்ள தனக்கு வேண்டிய அடிப்படை விவரங்களை எடுத்துக் கொண்டு ஒருவகைக் கணிப்பில் ஈடுபட்டுக் கொண்டி ருக்கும் போது, ஆய்வன் 2 அதே நினைவகத்திலிருந்து தனக்கு வேண்டிய விவரங்களை எடுத்துக் கொண்டு வேறு ஒரு கணிப்பைச் செய்யும். (இங்கு ஒன்றை நினைவு படுத்த விரும்புகிறேன். ஆய்வன் 1 நினைவகத்திலிருந்து தனக்கு வேண்டிய விவரங்களைச் சேகரித்துக் கொண்டிருக்கும் அதே வேளையில், ஆய்வன் 2 ம்,

நினைவகத்திலிருந்து தனக்கு வேண்டிய விவரங்களைச் சேகரிக்க முடியாது. ஆய்வின் 1, விவரங்களை நினைவகத்திலிருந்து எடுத்துக்கொண்டு கணிப்பிலீடுபட்டிருக்கும்போதுதான் ஆய்வன் 2 நினைவகத்தை நாடமுடியும். ஆக பல ஆய்வன்கள் இருப்பினும், நினைவகத்தை ஒரே சமயத்தில் ஒரு ஆய்வன்தான் நாடமுடியும்). வேறு விதமாகக் கூறவேண்டுமாயின் நினைவகம், ஆய்வன்களுக்கு ஒன்றன் பின் ஒன்ருகத்தான் அடிப்படை விவரங்களை அளிக்கும். ஒரே சமயத்தில் ஒன்றிற்கு மேற்பட்ட ஆய்வன்களை அது 'கவனிக்க' இயலாது. இத்தகைய கம்ப்யூட்டர் ஒருங்குகள் சில வற்றில் நினைவகங்களும் ஒன்றிற்கு மேற்பட்டு இருக்கலாம்; எல்லா நினைவகங்களையும் எல்லா ஆய்வன்களும் பயன்படுத்துமாறு இது அமைந்திருக்கும்.



படம் 7.2

பல — ஆய்வன் ஒருங்கின் இரண்டாம் வகை

இந்த அமைப்பில் ஒரு பெரிய சௌகரியம் உள்ளது. சில பல்லாய்வான் ஒருங்குகளில் ஒரு ஆய்வன் செயல்பட்டுக்கொண்டிருக்கும்போது திடீரென்று அதில் பழுது ஏற்பட்டுக் கணிப்பைத் தொடர இயலாது போதுமாயின், மற்ருெரு ஆய்வன் உடனே அக்கணிப்பை மேற்கொண்டு தொடர்ந்து செய்யும். இத்தகைய ஒரு வசதி விண்வெளிப் பயணங்களில் இன்றியமையாததாகும்.

நிகழ்நேர ஒருங்குகள் (Real Time Systems)

சாதாரணமாகக் கம்ப்யூட்டருக்கு நாம் ஒரு புரோகிராமைத் தயாரித்து அளித்தால், அந்த புரோகிராமை ஆரம்பித்துக்

கணித்து முடியும்வரைக் கம்ப்யூட்டரின் இயக்கத்தில் நாம் தலையிட முடியாது. கணிப்பினூடே நமக்கு ஏதாவது இடைமுடிவுகள் (Intermediate Results) தேவைப்படின் அவற்றை நமக்குக் கொடுக்கவேண்டி, உரிய ஆணைகளை முதலிலேயே நாம் புரோகிராமில் சேர்த்திருக்க வேண்டும்.

சில அமைப்புகளில் ஒரு குறிப்பிட்ட நிகழ்ச்சி நடக்க நடக்க அத்துடன் சம்பந்தப்பட்டக் கணிப்புகளை உடனுக்குடன் கம்ப்யூட்டர் செய்ய வசதியுண்டு. எடுத்துக்காட்டாக விமானப் பயண ரிசர்வேஷன் செய்யும் கம்ப்யூட்டரில், வாடிக்கையாளர்கள் வரும்போது மட்டும் கம்ப்யூட்டருக்கு இருக்கை ஒதுக்குவதற்கான ஆணை கொடுக்கப்படுகிறது. கொடுக்கப்பட்ட, ஆணைகளை உடனே நிறைவேற்றி உரிய இருக்கைகளை ஒதுக்குவதையோ, அல்லது ஏதும் காலியில்லை என்று அறிவிப்பதையோ செய்கிறது. இத்தகைய உபயோகத்தையே நிகழ்நேர உபயோகம் (Real Time Applications) என்பர். ஒரு கம்ப்யூட்டர் இத்தகைய பணி ஒன்றைச் செய்துகொண்டு இருக்குமாயின், அதை நிகழ்நேர கம்ப்யூட்டர் ஒருங்கு (Real Time Computer System) என்று அழைக்கிறார்கள். இத்தகைய கம்ப்யூட்டர்கள் முன் கூறியவாறு விமானப் பயணச் சீட்டு விற்பது மட்டுமல்லாமல் உற்பத்திக் கண்காணிப்பு (Process Control), பாதுகாப்புத் துறையில் விண்வெளிக்கணிப்பு (Air Space Surveillance) ஆகியவற்றிலும், இன்னும் பிற துறைகளிலும் பயன்படுத்தப்படுகிறது.

பெரும்பாலான நவீன கம்ப்யூட்டர்கள் நாம் சற்றுமுன்பார்த்த அம்சங்கள் பலவற்றையும் ஒருங்கே கொண்டதாய் அமைந்துள்ளன. ஒரு கம்ப்யூட்டரே பல புரோகிராம் கம்ப்யூட்டராகவும், பல் ஆய்வன் ஒருங்காகவும், நிகழ்நேர கணிப்புகளைச் செய்யக் கூடியதாயும் உற்பத்தி செய்கிறார்கள். இதனால் இன்றைய கம்ப்யூட்டர்களில் ஆற்றல்கள் விஸ்தரிக்கப்பட்டு அவற்றின் பயனும் பல மடங்காக்கப்பட்டுள்ளது.

BIBLIOGRAPHY

S. No.	Title	Publisher	Author
1.	Advances in Computers Vol. 1 to 5.	Allied Publishers	Rubinoff and Franz
2.	Automatic Data Processing	John Wiley & Sons	Iverson
3.	An Introduction to Computer Programming	Addison Wesley	Adolph C. Nydegger
4.	Basic Data Proceesing	Holt Reinhart Winston	Abrams and Corrime
5.	Business Date Processing-An Introduction	Addison Wesley	Wheeler - Jones
6.	Computers and their uses	Prentice Hall Series	Desmonde
7.	Computers and Education	Mc Graw Hill	Gerard
8.	Computers and Common sense	Columbia University Press	M. Taube
9.	Computers	Holt Library of Science	Shirley Thomas
10.	Digital Computers Principles	Van Nostrand	Irwin
11.	Digital Computer Fundamentals	Mc Graw-Hill	Thomas C. Bartee
12.	Electronic Data processing systems	Preutice Hall	O'Neil
13.	Exploring the Computer	Addison Wesley	Paul Allen
14.	Fortran Programming	Publishers - Merril	Dale Harris
15.	Introduction to Computers for Engineering	John Wiley	Moyle
16.	Man Machine Systems in Education	Harper and Row	Jon W-Loughary
17.	Man, Memory and Machines	Macmillan	Jacker
18.	The Language of Computers	Mc Graw Hill	Galler
19.	The Computer Age	Harper and Row	Burck
20.	Understanding Computers	Mc Graw Hill	Crowley

8 கலைச் சொற்கள்

Accuracy : துல்லியம்
Adding Machine : கூட்டல் பொறி
Alphabetic Sorting Switch . அகரப் பித்தான்
Alphameric Mode : எண் எழுத்து நிலை
Arithmetic Unit : கணிப்பங்கம்
Astronomical Table : விண் நிலைப் பட்டியல்
Binary : பைனரி; இருமை
Binary Coded Decimal Digit : இருமைக் குறி தசம எண்
Bit : 'பிட்'
Calculator : கணிப்பொறி, கால்குலேட்டர்
Card Code : கார்டுக் குறி
Card Punch : கார்டுத் துளைப் பொறி
Card Reader : கார்டு படிப் பொறி
Central Processing unit : மத்திய ஆய்வங்கம்
Character : உரு
Check Bit : சோதனை 'பிட்'
Coding Procedure : குறியீட்டு முறை; குறி முறை
Collator : கொலேட்டர்; இணைப்பி
Computation : கணிப்பு
Computer : கம்ப்யூட்டர், கணிப்பொறி, கணக்கி
Computer System : கம்ப்யூட்டர் ஒருங்கு
Control unit : ஆளங்கம்
Data : அடிப்படை விவரம்
Demagnetised State : காந்தமற்ற நிலை
Electric node : மின் முனை
Electric Pulse : மின் துடிப்பு
Electromechanical : மின்னியந்திர
Electronic : எலக்ட்ரானிக்

Flag Bit : கொடி 'பிட்'
Fortran Expression : போர்ட்ரான் தொடர்
Gang Punch : கூட்டுத் துளை
Hand Computation : கைக் கணிப்பு
High Density Mode : மிகை அடர்வு நிலை
High Speed . மிகை வேகம்
Higher Level Language : உயர்நிலை மொழி
Input/Output Media : கொள்/கொடு துறைகள்
Input unit : கொள்ளங்கம்
Instruction : ஆணை; கட்டளை
Integrated circuit : ஒன்றிய மின் வழி
Interpreter : விளக்கி
Key Board : பித்தான் முகம்
Light pen : ஒளிப் பேனா
Low Density Mode : குறை அடர்வு முறை
Machine Language : பொறி மொழி
Macro Instruction : பேராணைகள்
Magnetic Disc (Disk) : காந்தத் தட்டு
Magnetic Tape : காந்த நாடா
Magnetised State : காந்தமுற்ற நிலை
Mark Sense Card : குறியுணர் கார்டு
Matching : இணைத்தல்
Memory : நினைவகம்
Memory Cell : நினைவறை
Micro Instruction : சிற்றாணை
Micro Miniaturisation : நுண் சுருக்கம்
Merging : ஒன்றல்
Numeric Bit : எண் 'பிட்'
Numeric Mode : எண் நிலை
Odd Parity Check : ஒற்றைச்சமன் சோதனை
Off State : அணை நிலை
On Line : நேர்முக
On State : எரி நிலை
Operation Code , செயல் குறி
Operation Research : ஆய்வுப் பாடு
Output Unit : கொடங்கம்
Panel : பொறிப் பலகை
Paper Tape : காகித நாடா
Port-a-punch Card : உடனடிக் துளைக் கார்டு
Problem : பிரச்சினை; கணக்கு

Processing	:	ஆய்தல்
Program	:	புரோகிராம்; கட்டளைக்கொத்து
Punch	:	துளை
Punched Card	:	துளைக் கார்டு
Punching Die	:	துளை அச்சு
Punching Machine	:	துளைப் பொறி
Random Access	:	கட்டிலாக் கொள் முறை
Reproducer	:	விருத்தி
Reading Station	:	படிப்பிடம்
Read/Write Head	:	படி/எழுது முனை
Sector	:	துண்டம்
Sector Address	:	துண்ட முகவரி
Selecting	:	தேர்ந்தெடுத்தல்
Select and Merge	:	தேர்ந்தொன்றல்
Sensor	:	உணர் முனை
Sequence Check	:	வரிசைச் சோதிப்பு
Simulation	:	போலி நிகழ்வு
Space Bar	:	வெளிச் சட்டம்
Special Character	:	தனிக் குறி; சிறப்புக் குறி
Sorter	:	வகைப்படுத்தி, முறைப் பொறி; சார்ட்டர்
Stored Program Computer	:	கட்டளைக் கொத்து (புரோகிராம்) ஏற்கும் கம்ப்யூட்டர்
Summary Report	:	சுருக்க அறிக்கை