INTERNATIONAL ATOMIC ENERGY AGENCY
UNITED NATIONS EDUCATIONAL, SCIENTIFIC AND CULTURAL ORGANIZATION

INTERNATIONAL CENTRE FOR THEORETICAL PHYSICS
34100 TRIESTE (ITALY) - P.O.B. 586 - MIRAMARE - STRADA COSTIERA 11 - TELEPHONES: 224281/2/3/4/5/6
CABLE: CENTRATOM - TELEX 460392-I

SMR/90 - 9

COLLEGE ON MICROPROCESSORS:

TECHNOLOGY AND APPLICATIONS IN PHYSICS

7 September - 2 October 1981

MICROPROCESSOR ARCHITECTURE

D. ASPINALL

Dept. of Computation
U.M.I.S.T.
Manchester
England

1

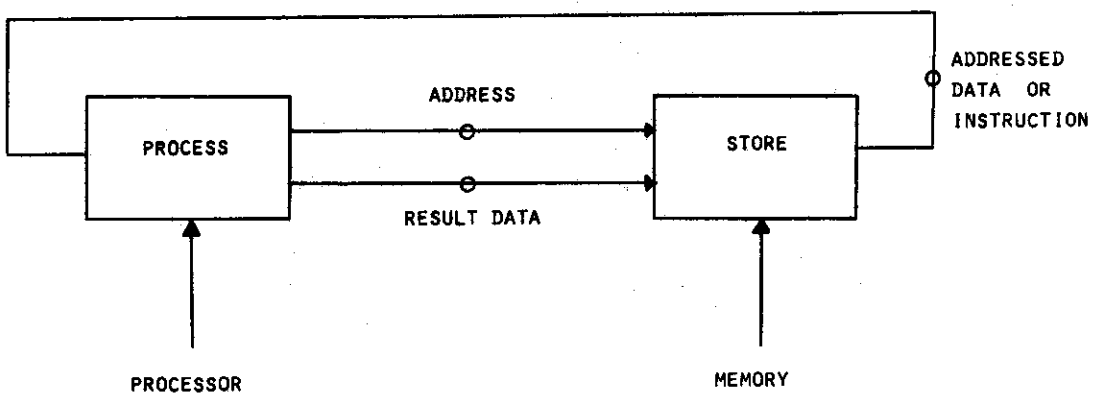


FIG. 1 : INSTRUCTION SET PROCESSOR-MEMORY PAIR

2

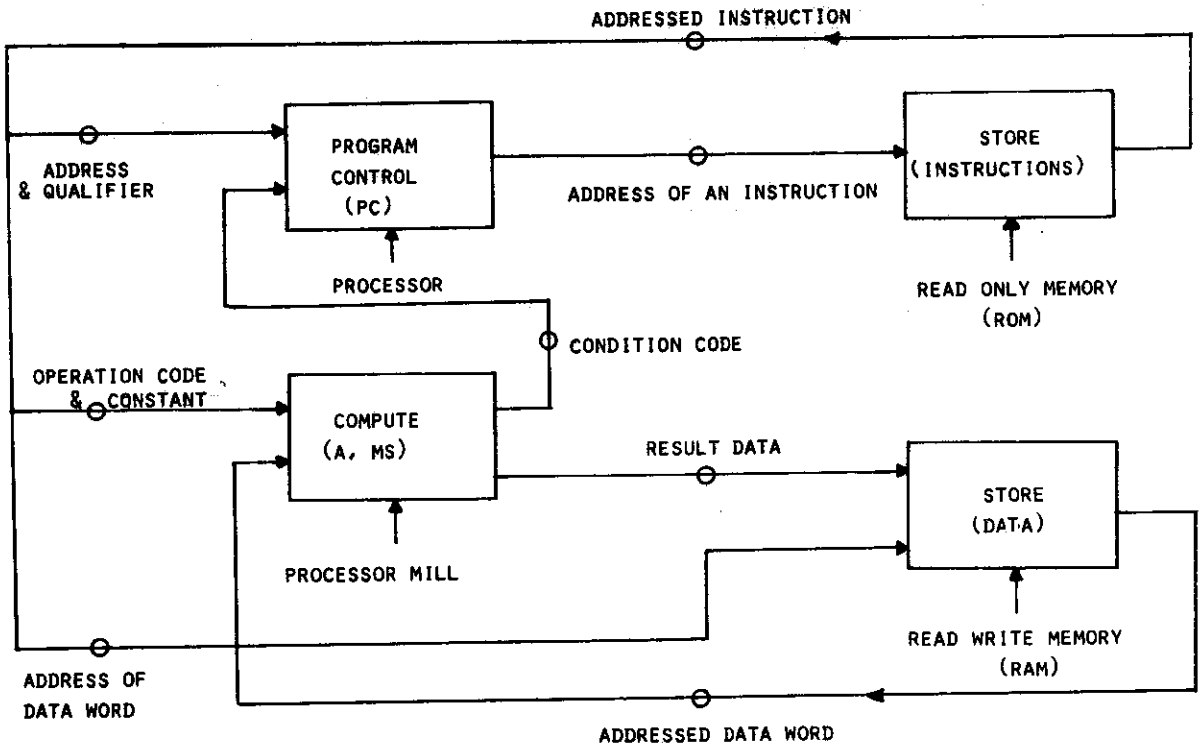


FIG. 2 : EARLY COMPUTER (HARVARD)

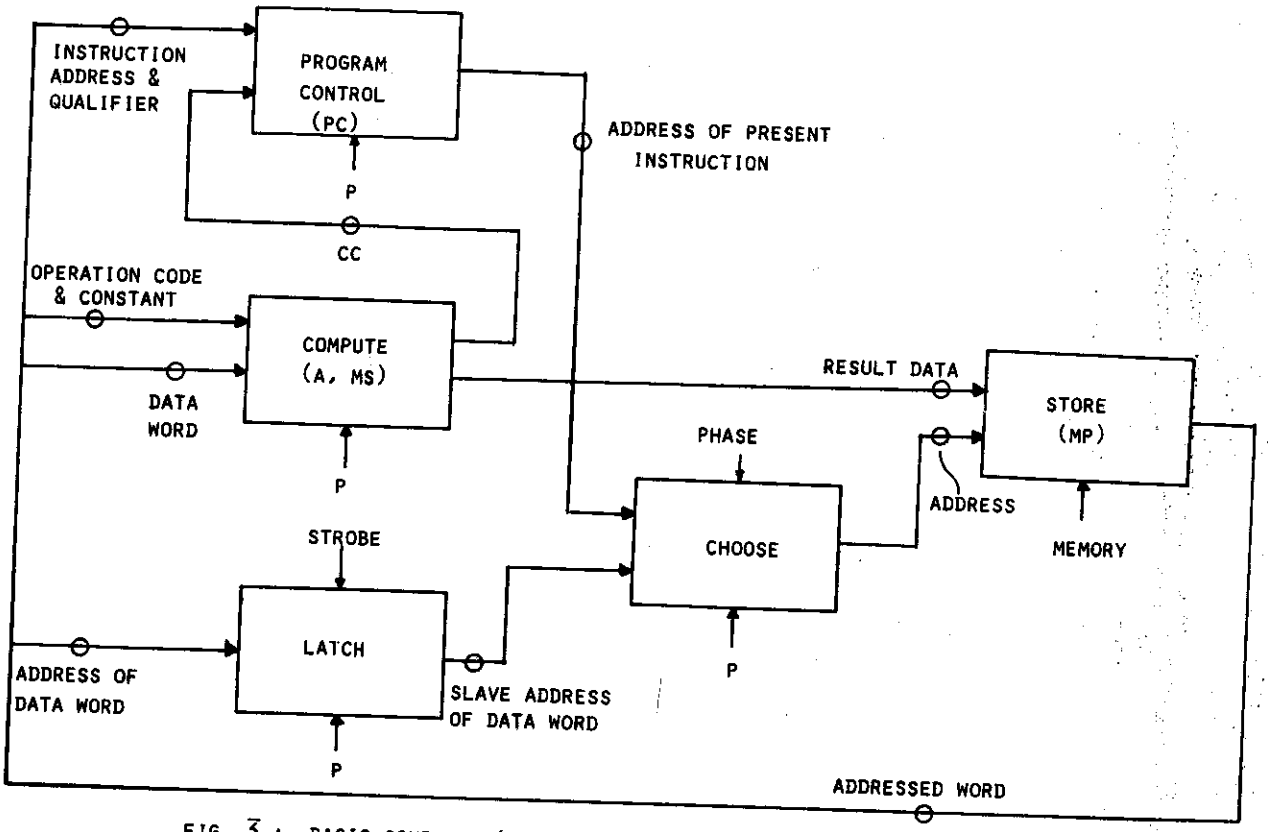


FIG. 3 : BASIC COMPUTER (MOORE-VON NEUMAN)

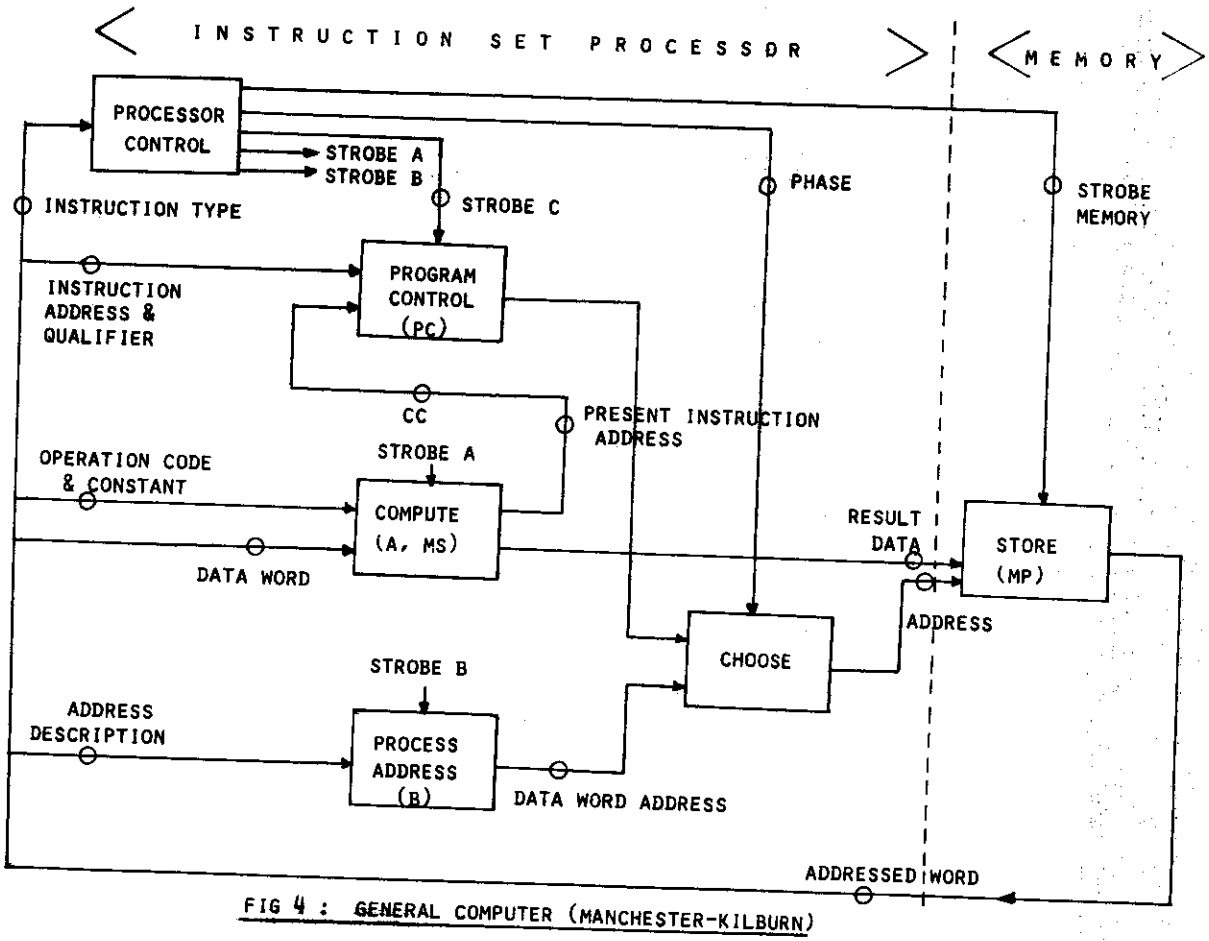


FIG 4 : GENERAL COMPUTER (MANCHESTER-KILBURN)

ADDRESSING SCHEME	ASSIGNMENT STATEMENT	
	INSTRUCTION	MACHINE ACTION
THREE-ADDRESS	Mx := My b Mz	As INSTRUCTION
TWO-ADDRESS	Mx := My b Mx	As INSTRUCTION
ONE-ADDRESS	b Mx STORE Mx	A := A b Mx Mx := A
STACK (ZERO-ADDRESS)	b PUSH, Mx POP	ST := ST b STN ST := Mx M(STN) := ST

$M_{a\backslash}$ Value of Data Word at Memory Location a

a\x\y\z\(STN)

b\ Binary Operator A\ Accumulator

ST\ Value of Data Word at Top of Stack

STN\ Value of Data Word at Next below Top of Stack

(STN)\ Memory location given by STN

TABLE 1 : ADDRESSING SCHEMES

ADDRESSING SCHEME	PDP-11	68000	8086	Z-8000
TWO-ADDRESS	✓	-	-	-
ONE-ADDRESS (GENERAL REGISTER)	✓	✓	✓	✓
ADDRESSING MODE				
LITERAL	✓	✓	✓	✓
DIRECT	✓	✓	✓	✓
IN-DIRECT	✓	✓	✓	✓
AUTO-INCREMENT (POST)	✓	✓	-	-
AUTO-DECREMENT (PRE)	✓	✓	-	-
REGISTER LITERAL	✓	✓	✓	✓
REGISTER-REGISTER	-	-	✓	✓
REG, -REG, -LITERAL	-	✓	✓	-
POST-INC, DEFERRED	✓	-	-	-
PRE-DEC, DEFERRED	✓	-	-	-
REGISTER-LITERAL DEFERRED	✓	-	-	-
MAXIMUM ADDRESSABLE MEMORY SIZE (BYTES)	64K	16M	1M	8M

TABLE 2 : PHYSICAL ADDRESSING SCHEMES AND MODES

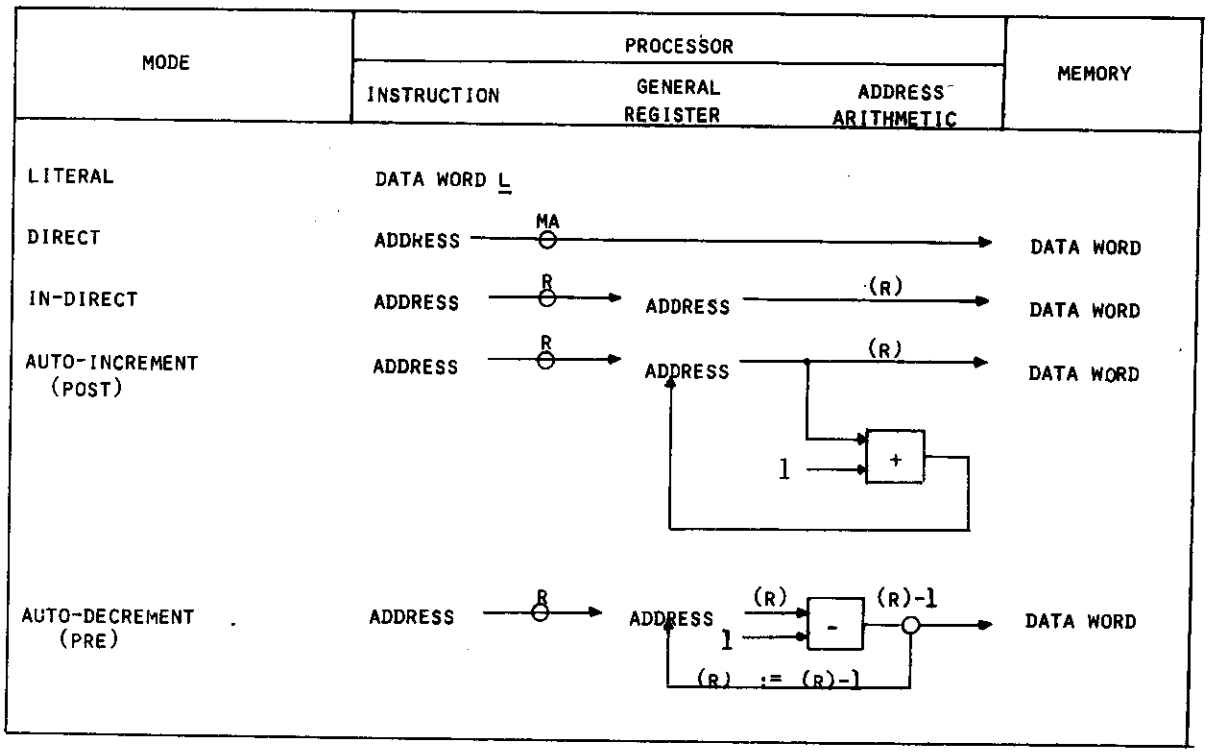


FIGURE 1(A) : MICRO-PROCESSOR ADDRESSING MODES

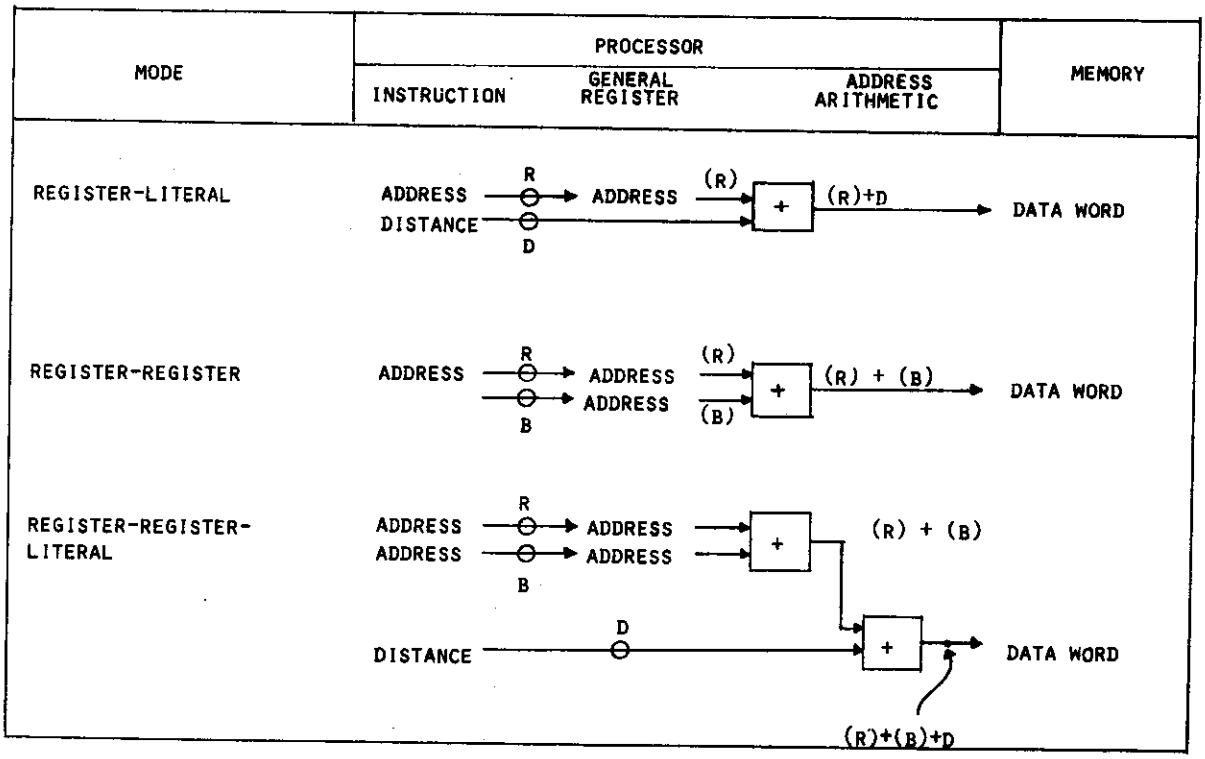


FIGURE 1(B) : MICRO-PROCESSOR ADDRESSING MODES

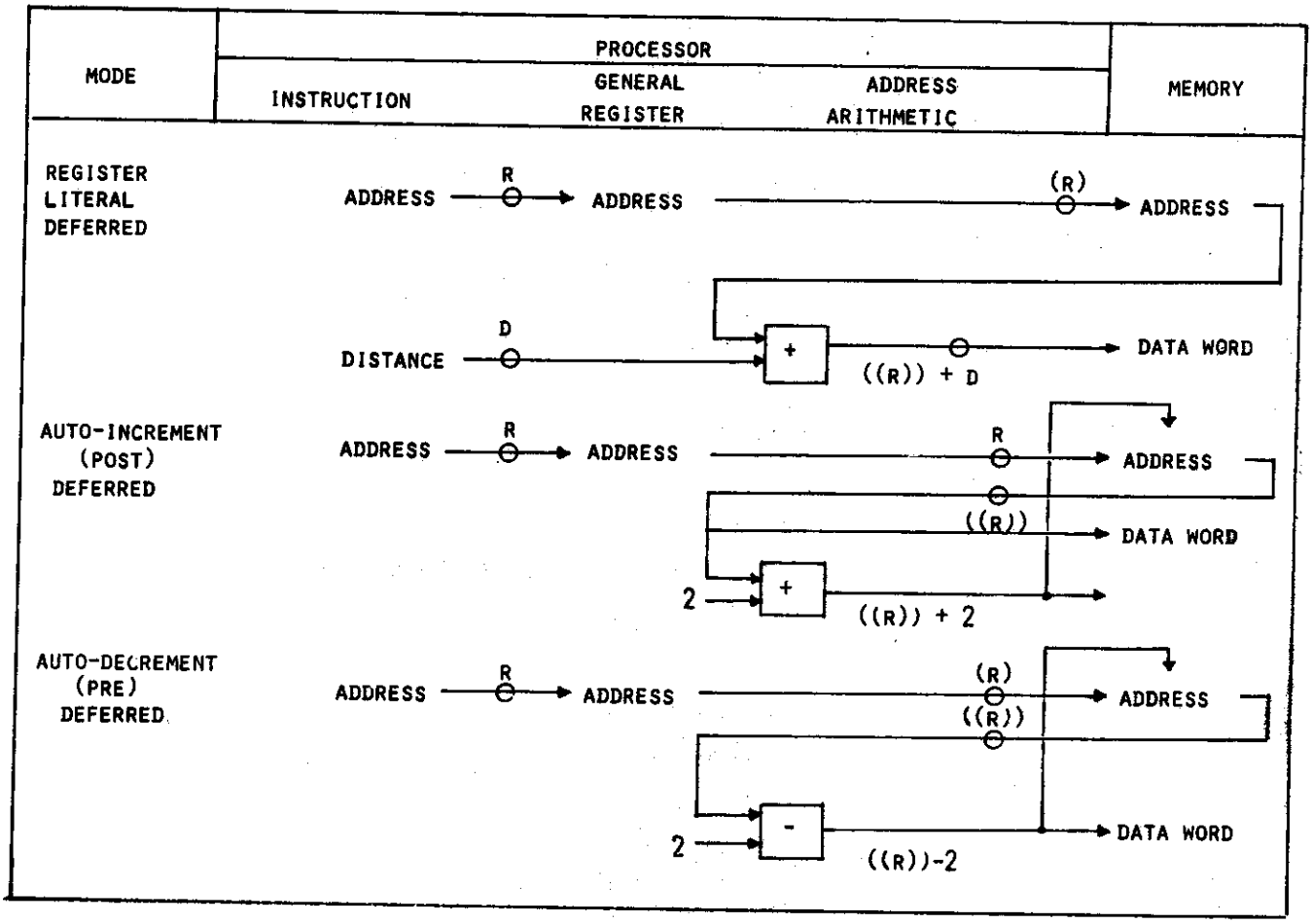


FIG. 2 : PDP-11 DEFERRED ADDRESSING MODES

10

ADDRESSING MODES

LITERAL DATA WORD = L

DIRECT DATA WORD = (MA)

IN-DIRECT REGISTER VALUE = (R) NEXT DATA WORD = ((R))

ADDRESSING MODES

AUTO-INCREMENT (POST)

REGISTER VALUE = (R)
NEXT DATA WORD = ((R))
NEXT (R) := (R) + 1

AUTO-DECREMENT (PRE)

REGISTER VALUE = (R)
NEXT (R) := (R) - 1
NEXT DATA WORD = ((R))

ADDRESSING MODES

REGISTER-LITERAL

REGISTER VALUE: (R)
NEXT DATA WORD = ((R)) + D

REGISTER-REGISTER

GENERAL REGISTER VALUE = (R)
GENERAL REGISTER VALUE = (B)
NEXT DATA WORD = (R) + (B))

REGISTER-REGISTER-LITERAL

GENERAL REGISTER VALUE = (R)
GENERAL REGISTER VALUE = (B)
NEXT DATA WORD = ((CR) + (B) + D)

BINARY OPERATOR	
ADD	A := A + Ma
SUBTRACT	A := A - Ma
DIVIDE	A := A / Ma
MULTIPLY	A := A * Ma
AND	A := A $\wedge$ Ma
OR	A := A $\vee$ Ma

TABLE 3 : AVAILABLE OPERATIONS

THREE-ADDRESS MACHINE ON ONE-ADDRESS MACHINE

THE THREE-ADDRESS SCHEME, Mx := My b Mz, MAY BE MODELLED ONTO A ONE-ADDRESS MACHINE AS SHOWN :

E.G. Mx := My + Mz

LOAD My	A := My
ADD Mz	A := A + Mz
STORE Mx	Mx := A

GENERAL REGISTERS :

A	TOP OF STACK
B	TEMPORARY STORE
SP	STACK POINTER

STATEMENT b * OPERATE *

```
A := A b (SP)  !  INDIRECT MODE
SP := SP + 1   !  INCREMENT GENERAL REGISTER
```

STATEMENT PUSH Mx *PUSH DATA WORD FROM LOCATION x ONTO STACK*

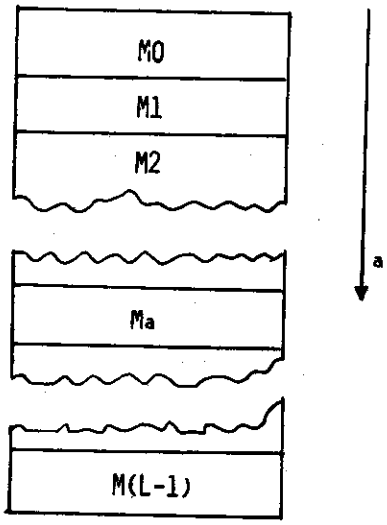
```
SP := SP - 1   !  DECREMENT GENERAL REGISTER
(SP) := A      !  INDIRECT MODE
A := Mx        !  Mx BY ANY MODE
```

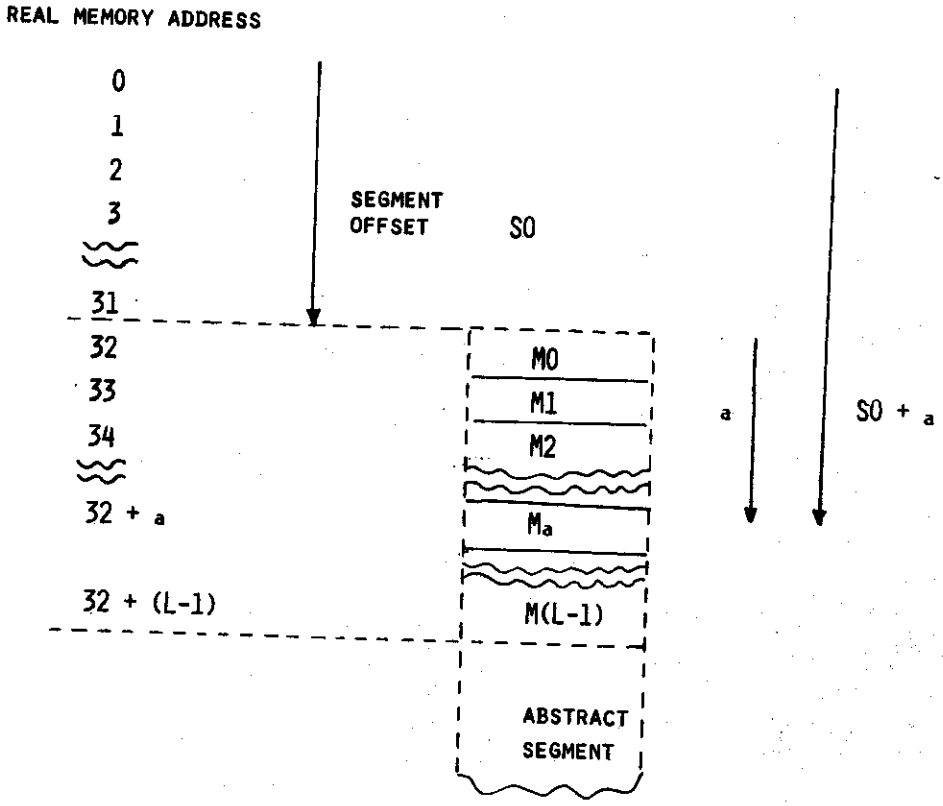
STATEMENT POP *POP WORD FROM TOP OF STACK TO LOCATION GIVEN BY ADDRESS HELD IN NEXT TO TOP OF STACK*

```
B := (SP)      !  MOVE ADDRESS INTO GENERAL REGISTER B
SP := SP + 1
(B) := A       !  STORE A AT ADDRESS GIVEN BY B
A := (SP)      !  MOVE NEW TOP OF STACK TO A
SP := AP + 1   !  INCREMENT POINTER
```

```
b
   A  :=  A b (SP)    NEXT  SP  :=  SP + 1     !  POST AUTO-INC.

PUSH  Mx
   SP  :=  SP - 1     NEXT  (SP)  :=  A        !  PRE AUTO-DEC.
   A  :=  Mx

POP
   B  :=  (SP)        NEXT  SP  :=  SP + 1
   (B)  :=  A
   A  :=  (SP)        NEXT  SP  :=  SP + 1
```

```
b
   POP D                          !  OPERAND AT NEXT TO TOP OF STACK
                                     MOVED TO GENERAL REGISTER D
   A  :=  A b D

PUSH  Mx
   PUSH A                         !  ACCUMULATOR VALUE PUSHED TO NEXT TO
                                     TOP OF STACK
   A  :=  Mx

POP
   POP B
   (B) :=  A
   POP  A
```


M0
M1
M2
a
Ma
M(L-1)


REAL MEMORY ADDRESS
0
1
2
3
31
32
33
34
32 + a
32 + (L-1)
SEGMENT OFFSET
S0
M0
M1
M2
Ma
M(L-1)
ABSTRACT SEGMENT
a
S0 + a

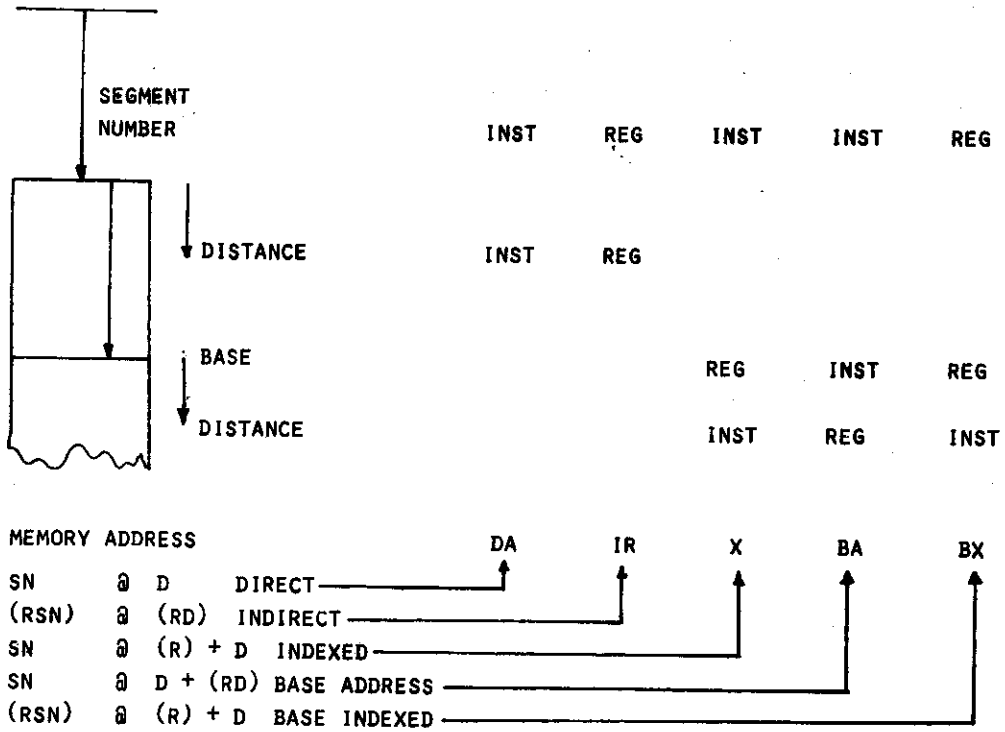


FIG. 3 : Z-8000 SEGMENTED ADDRESSING

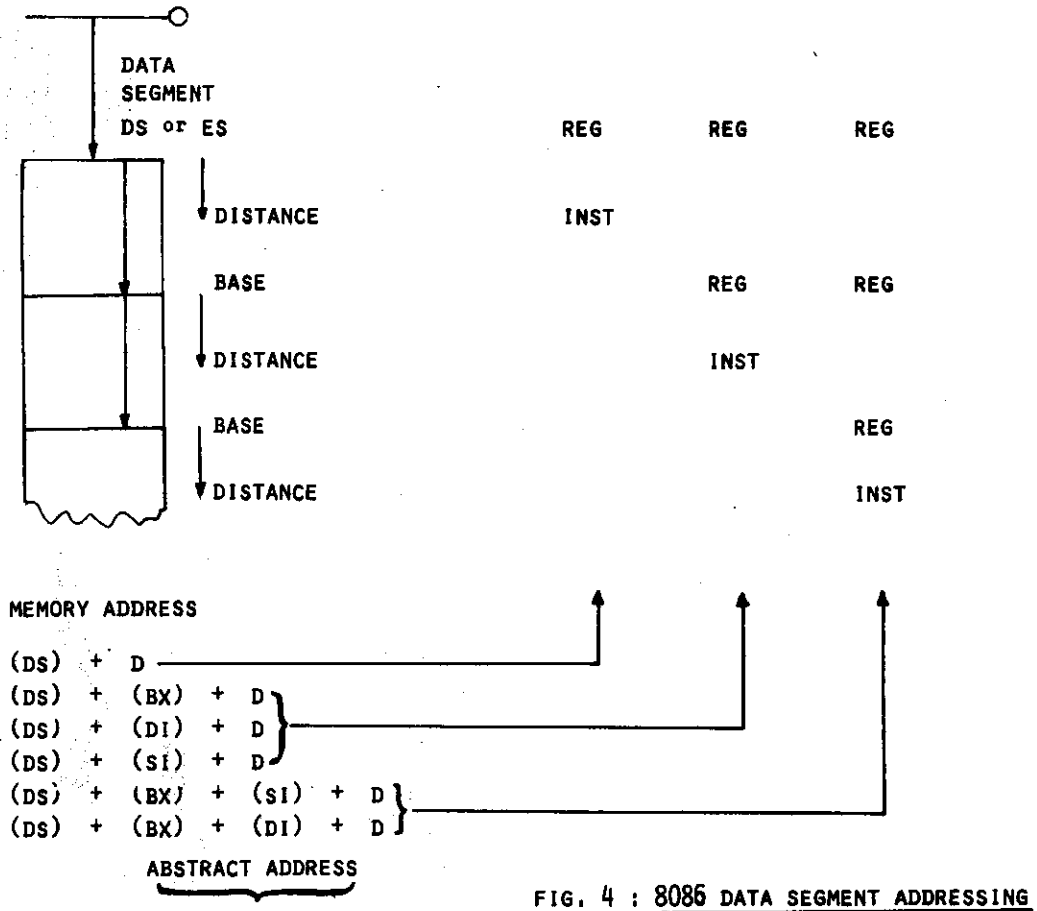


FIG. 4 : 8086 DATA SEGMENT ADDRESSING

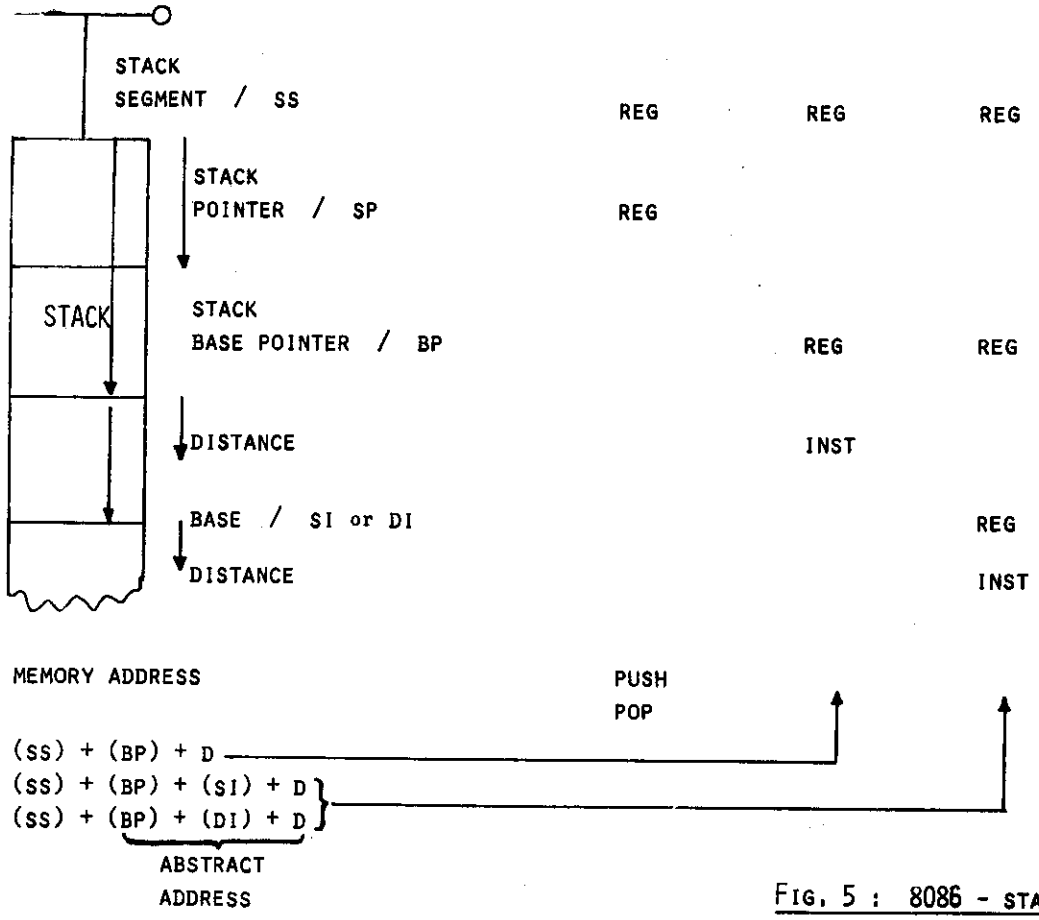
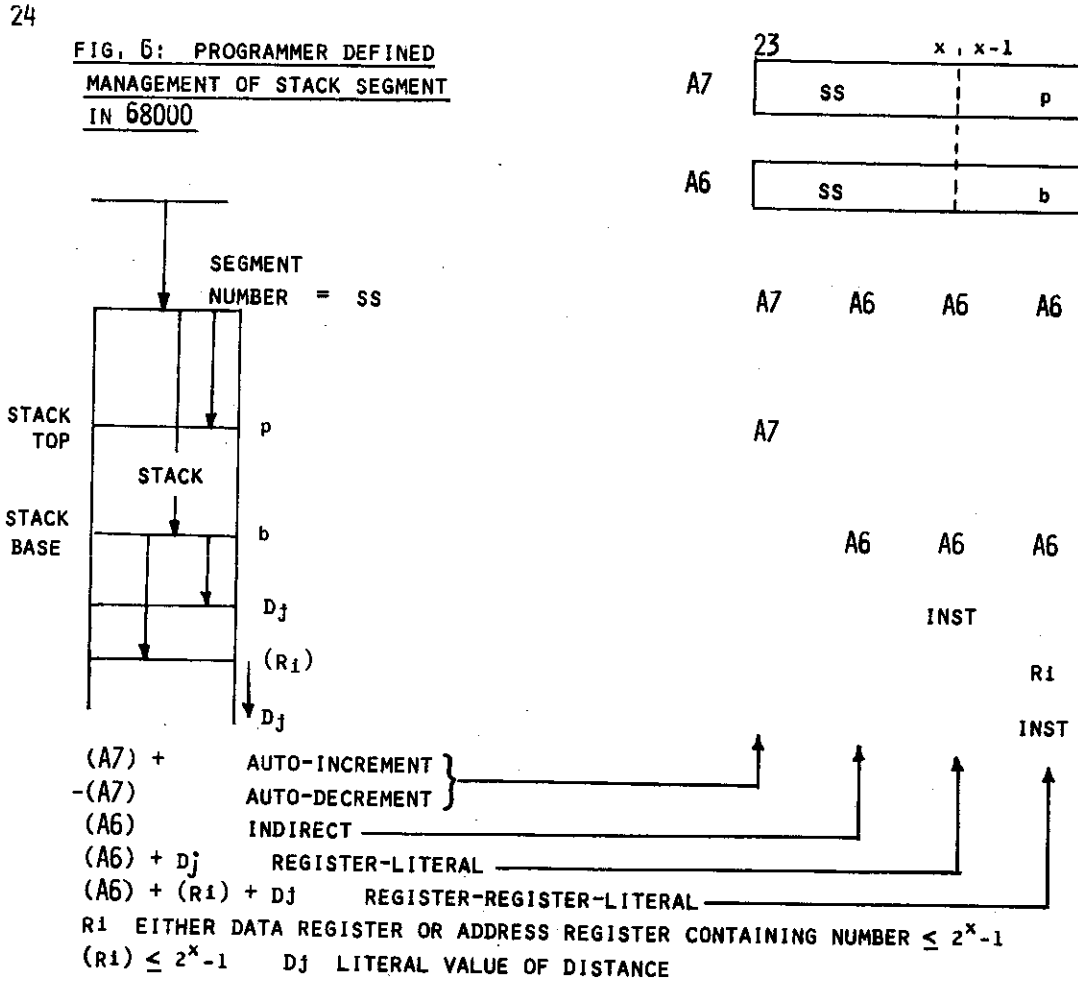


FIG. 5 : 8086 - STACK SEGMENT ADDRESSING

24

FIG. 6: PROGRAMMER DEFINED MANAGEMENT OF STACK SEGMENT IN 68000



Ri EITHER DATA REGISTER OR ADDRESS REGISTER CONTAINING NUMBER $\leq 2^x-1$

$(Ri) \leq 2^x-1$ Dj LITERAL VALUE OF DISTANCE

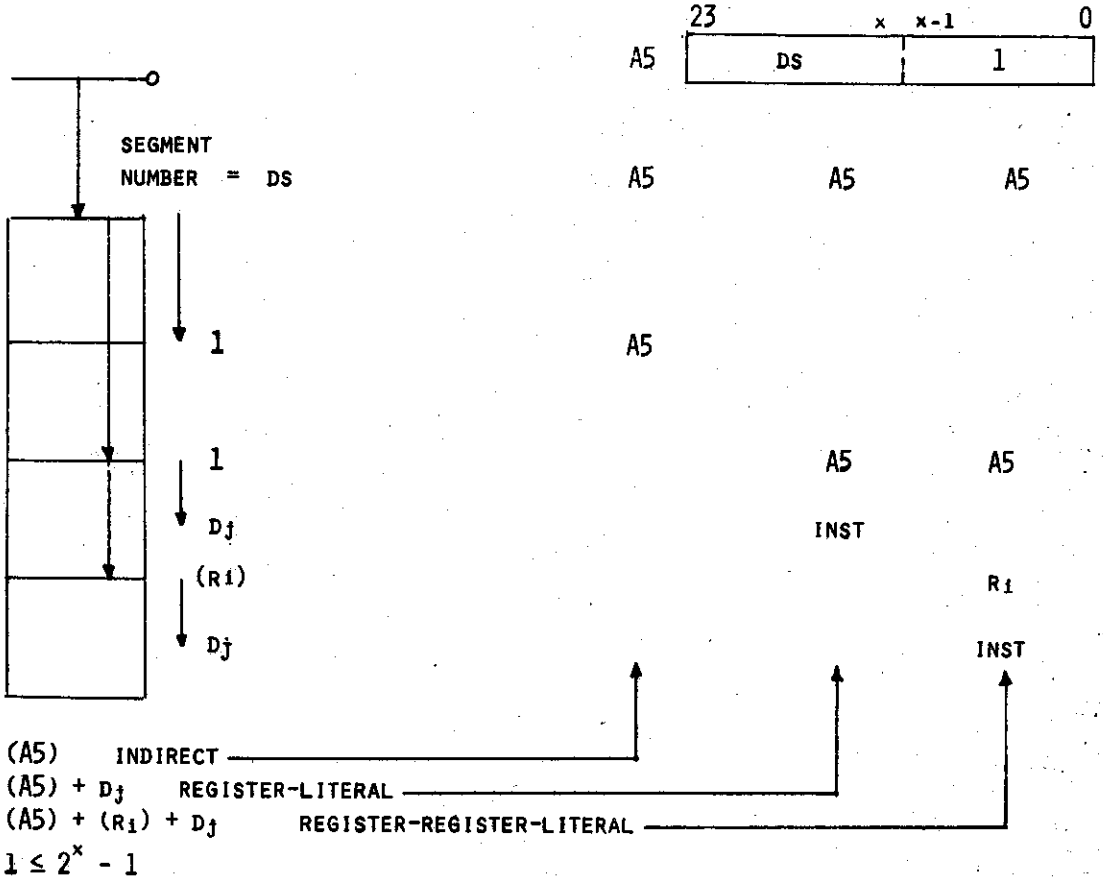


FIG. 7 : PROGRAMMER DEFINED MANAGEMENT OF DATA SEGMENT IN 68000

SELECTION CLAUSE

if Q then PC := PC + ℓ else PC := PC + 1

REPETITIVE CLAUSES

Repeat A until Q ,

& While Q do A

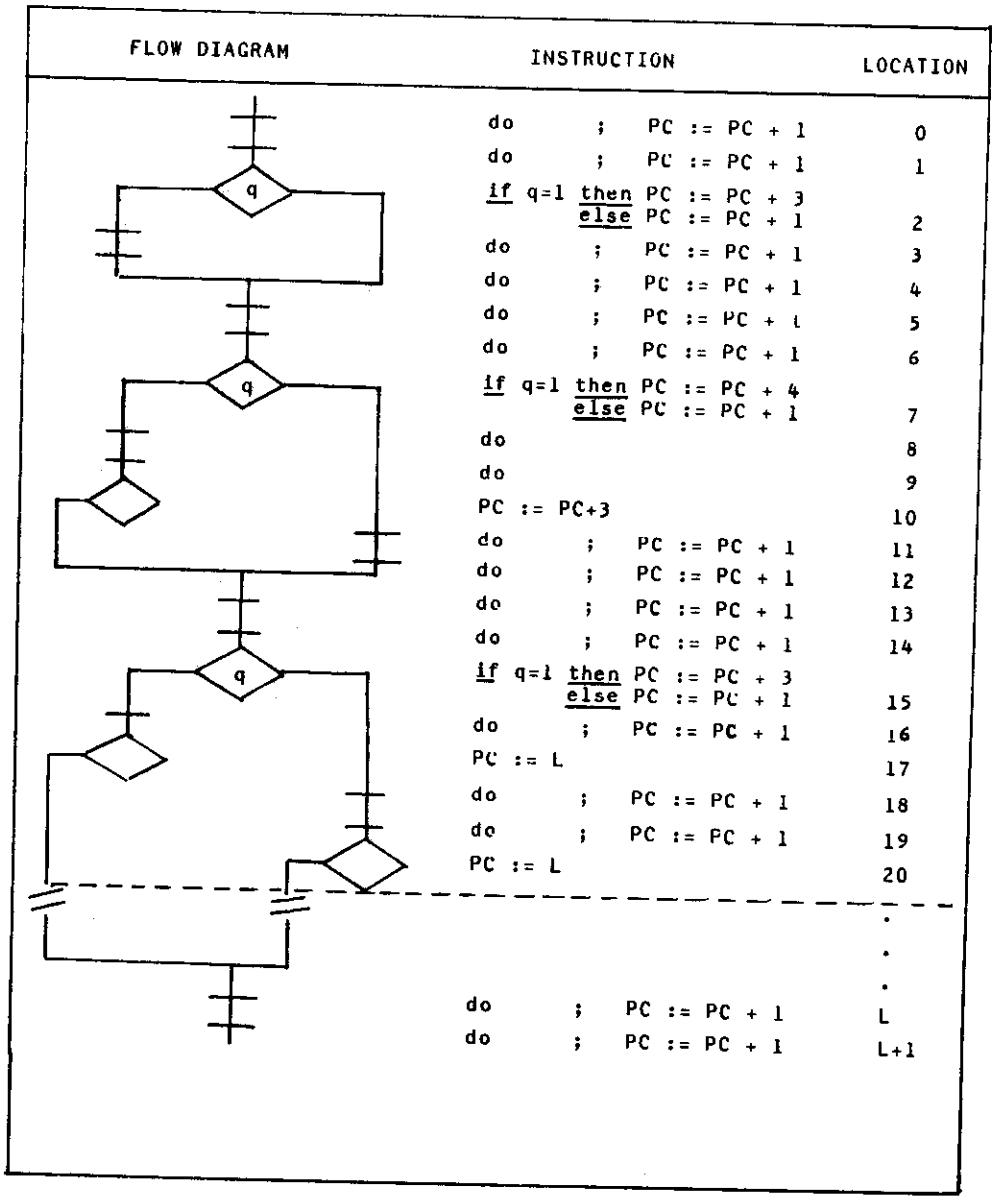


FIG. 1 : SELECTION CLAUSE PROGRAMME CONTROL

8086 LOOP

CX := CX - 1

if CX ≠ 0 then PC := PC + D else PC := PC + 1

$-128 \leq D \leq +127$

Z-8000 DJNZ

R1 := R1 - 1

if R1 ≠ 0 then PC := (PC + 1) + D else PC := PC + 1

$0 \leq D \leq 127$

8086 LOOP△

```
CX := CX - 1

if (△ = 1) & CX ≠ 0 then PC := PC + D else PC := PC + 1

- 128 ≤ D & ≤ + 127
```

MC 68000 DBCC

```
if CC = 1 then Dn := Dn - 1

                NEXT if Dn ≠ 1 then PC := PC + D
           else PC := PC + 1
-2^15 ≤ D ≤ 2^15 - 1 bytes
```

SUBROUTINE ENTRY AND EXIT

ENTRY :

```
PC := PC + 1      ! INCREMENT PROGRAMME COUNTER
DUMP := PC        ! DUMP CONTENTS OF PROGRAMME COUNTER
PC := ASR         ! LOAD PROGRAMME COUNTER WITH ASR
```

EXIT :

```
PC := DUMP        ! RESTORE RETURN ADDRESS TO PC
```

STACK OF DUMP LOCATIONS

CALLING ACTION :

```
PC  := PC + 1
SP  := SP - 1      } ! PUSH CONTENTS OF PC ONTO STACK
(SP) := PC
PC  := ASR
```

RETURN :

```
PC  := (SP)        } ! POP RETURN ADDRESS FROM STACK TO PC
SP  := SP + 1
```


	CHANGE TO PROGRAMME COUNTER	CONDITION FOR CHANGE					
		STATUS	COUNT	PDP-11	68000	8086	Z-8000
SELECTION CLAUSE	RELATIVE	✓	-	BΔ	BCC	JΔ	JRΔ
		-	-	BR	BRA	JMP*	-
	ABSOLUTE	✓	-	-	-		JPΔ
		-	-	JMP	JMP	JMP*	-
REPETITIVE CLAUSE	RELATIVE	-	✓	-	-	LOOP JCXZ	DJNZ
		✓	✓	-	DBCC	LOOPΔ	-
SUBROUTINE ENTRY	ABSOLUTE	-	-	JSR	JSR	CALL*	CALL
	RELATIVE	-	-	-	BSR	CALL*	CALR
SUBROUTINE EXIT	STACK	✓	-	-	-	-	RET
		-	-	RTS	RTS	RET	-

Δ :- ONE OF SEVERAL CONDITIONS OF MILL STATUS

* DEPENDS UPON ADDRESSING MODE

TABLE 1 : PROGRAMME CONTROL INSTRUCTIONS

ACTION ON A TRAP

```
SP   := SP - 1      } ! PUSH PROCESSOR STATUS WORD
(SP) := PSW         }
SP   := SP - 1      } ! PUSH RETURN ADDRESS
(SP) := PC          }
PC   := X           ! START ADDRESS OF SYSTEM ROUTINE
```

33

ACTION TO RESTORE

```
PC   := (SP)        } ! RESTORE RETURN ADDRESS
SP   := SP + 1      }
PSW  := CSP         } ! RESTORE PROCESSOR STATUS WORD
SP   := SP - 1      }
```

PROCESSOR	INSTRUCTION
PDP-11	RTI
68000	RTE (PRIVILEGED)
8086	IRET
Z-8000	IRET (PRIVILEGED)

34

PROCESSOR STATE VECTOR	PDP-11		M-68000		8086		Z-8000	
	REG	PSW	REG	PSW	REG	PSW	REG	PSW
MILL								
GENERAL REGISTERS - DATA	Rn		Dn		DX		Rn	
. ACCUMULATOR TOP OF STACK	Rn		Dn		AX		Rn	
. MILL STATUS		CC		CCR		FLAG		FLAG
. SIGN		N		N		S		S
. ZERO		Z		Z		Z		Z
. OVERFLOW		V		V		O		P/V
. CARRY								
32 BIT		-				-		
16 BIT		C		C		C		C
8 BIT		-				-		
4 BIT		-				A		
. EXTEND		-		X		-		-
. PARITY		-				P		P/V
. DECIMAL		-						DA
OPERAND ADDRESSING UNIT								
GENERAL REGISTERS -ADDRESS	Rn		An		BX		Rn	
. INDEX	Rn		An		SI,DI		Rn	
. STACK TOP POINTER	R6		A7		SP		R15	
. STACK BASE POINTER	Rn		An		BP		Rn	
. STATUS		-		-		D		-
PROGRAM CONTROL UNIT								
. PROGRAMME COUNTER	R7		PC		IP		PC	
. LOOP COUNTER	Rn		Dn		CX		Rn	
MEMORY MANAGEMENT UNIT								
. CODE SEGMENT REG.	R7		PC		CS		PC	
. STACK SEGMENT REG.	R6		A7		SS		R15	
. DATA SEGMENT REG.	Rn		An		DS		Rn	
. EXTRA DATA SEGMENT REG.	Rn		An		ES		Rn	
. STATUS		-		-		-		SEG
PROCESS CONTROL UNIT								
. CONTROL STATUS								
. PRIVILEGE		-		S		-		S/N
. TRACE		T		T		T		SE
. INTERRUPT LEVEL		P_{012}		I_{012}		I		VIE NVIE

PSW := PROCESSOR STATUS WORD

TABLE 2 : PROCESSOR STATE VECTOR

VECTOR NUMBER	68000	8086
0	RESET SSP	DIVIDE BY ZERO
1	RESET PC	TRACE
2	BUS ERROR	NMI (PIN 17)
3	ADDRESS ERROR	INT (TYPE 3)*
4	ILLEGAL INSTRUCTION	INT 0
5	DIVIDE BY ZERO	
6	CHK	
7	TRAPV	
8	PRIVILEGE VIOLATION	RESERVED
9	TRACE	
10	RESERVED	
31		
32	TRAP	INTR (PIN 18)
47		
48	RESERVED	AVAILABLE
63		
64	AVAILABLE	
255		

* INT $\overline{\text{(TYPE 3)}}$ CAN SPECIFY ANY VECTOR NUMBER 0 → 255

TABLE 3 : TRAP AND INTERRUPT VECTORS

37

HIGHEST PRIORITY	-	RESET
		BUS ERROR
		ADDRESS ERROR
		TRACE
		INTERRUPT REQUEST
		ILLEGAL INSTRUCTION
		PRIVILEGE VIOLATION
		TRAP, TRAPV, CHK
		DIVIDE BY ZERO
LOWEST PRIORITY	-	NORMAL INSTRUCTION

TABLE 4 : PRIORITY ASSIGNMENT WITHIN 68000

38

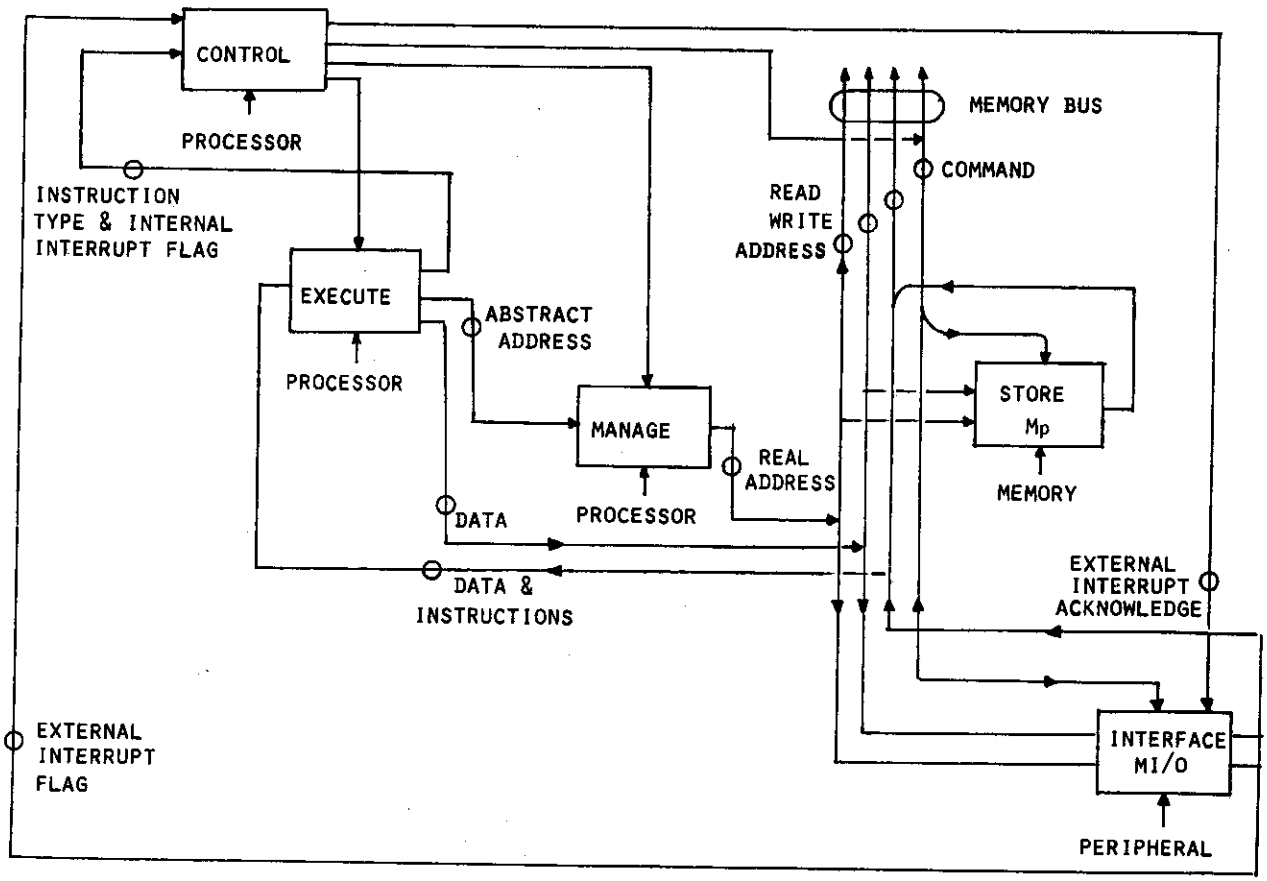


FIG. 1 : OVERVIEW OF MICROCOMPUTER SYSTEM

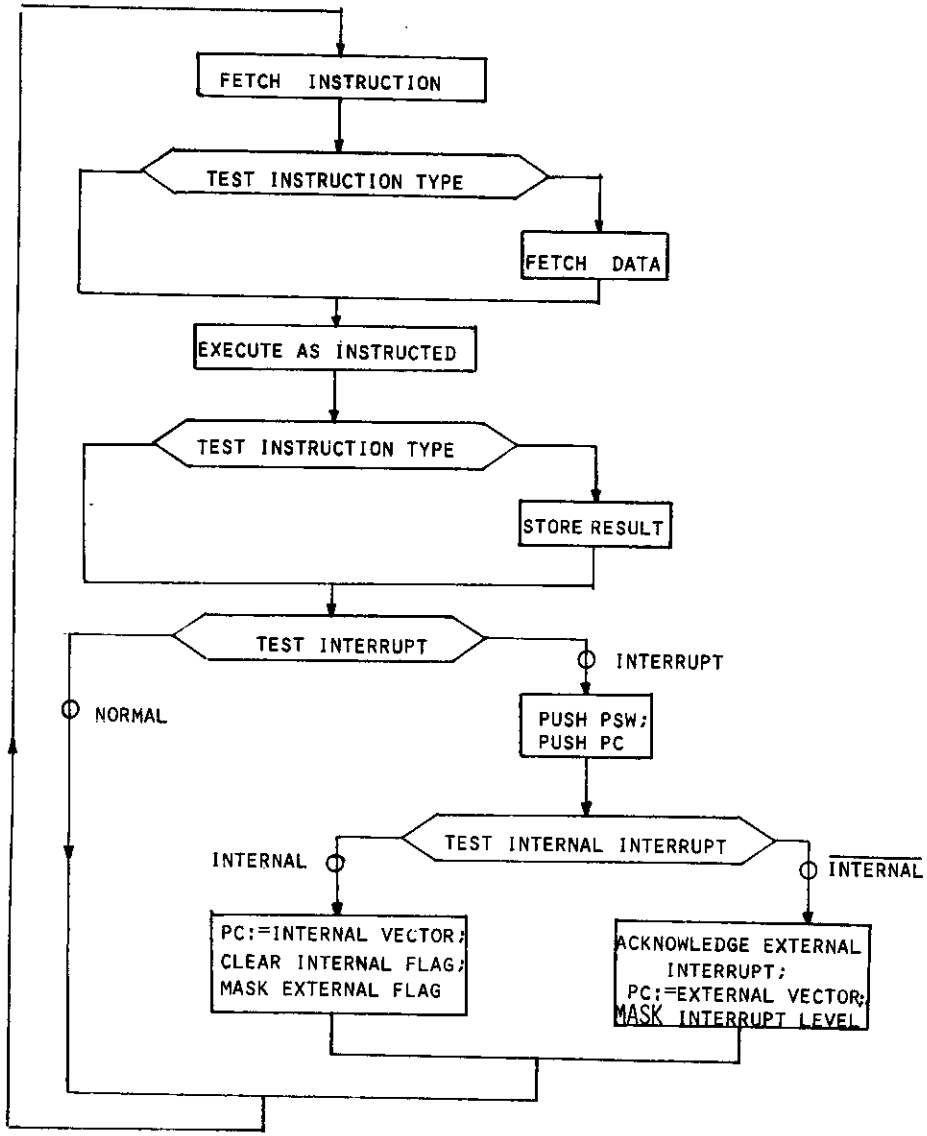


FIG 2 : FLOW DIAGRAM OF CONTROL SEQUENCE

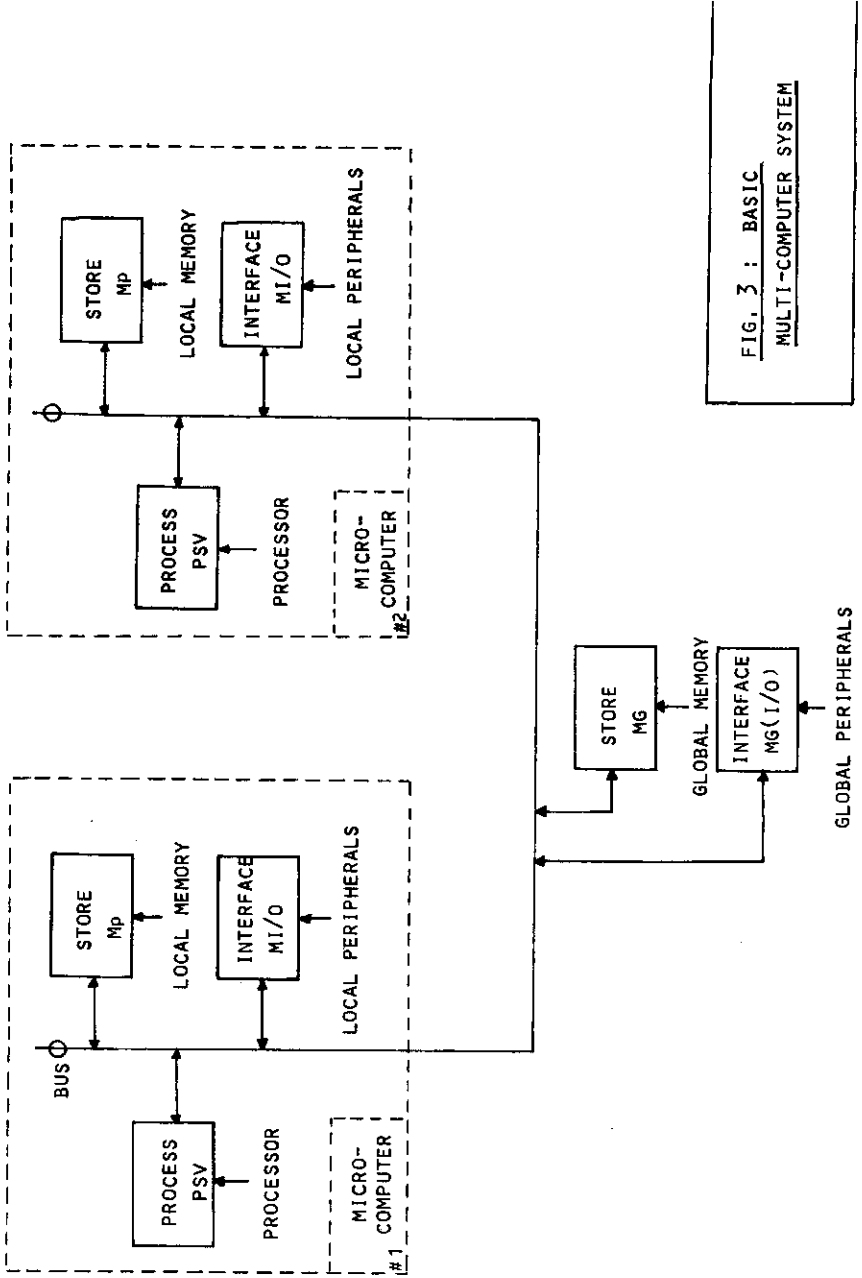


FIG. 3 : BASIC MULTI-COMPUTER SYSTEM