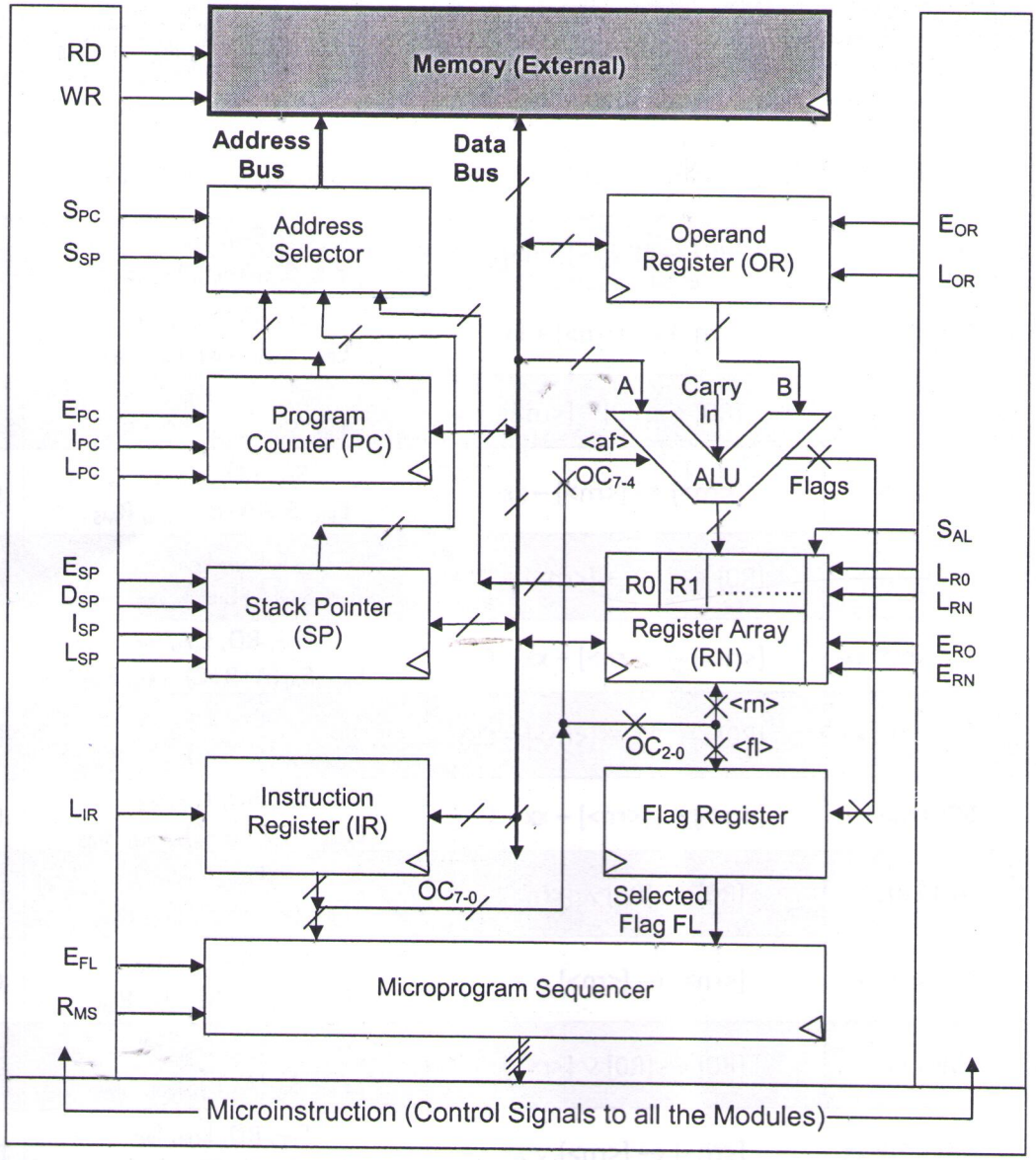
Single Bus Architecture



Nomenclature:

E_{XY} / RD : Enables the Output of the module XY/Memory on to the Data Bus;
L_{XY} / WR : Loads the Data Input applied to module XY/Memory ***at the next Active Clock edge***;
I_{XY} / D_{XY} : Increments/Decrements the value stored in module XY;
S_{PC} / S_{SP} :Selects the PC/SP output as the Memory Address (both = 0 selects R0);
R_{MS} :Resets the Microprogram Sequencer, subject to the values of **E_{FL}** and **FL**.
S_{AL} : Selects the ALU output as the Data Input to the Register Array;

The ALU has **16** functions (4-bit Select provided by the leading 4 bits of the Op Code):

Unary **8** [Codes 0 – 7]: 0, B, B', A, A', A + 1, A – 1, $2^n * A$ (Shift Left by ***n*** bits, C ← MSB after the shift).

Arithmetic **4** [Codes 8 – B]: A + B, A – B, A + B + C, A – B – C [Codes 8 – B].

Logic **4** [Codes C – F]: A AND B, A OR B, A XOR B, (A XOR B)' [Codes C – F];

The ALU generates **4** flags – Zero (Z), Carry (C), Sign (S), Parity (P). Flags are not affected by the Unary Logic functions. Only the **C** flag is affected by the Shift function. **All** flags are affected by the other ALU functions.

The Microprogram Sequencer is Reset ***at the next Active Clock edge*** **if $(FL' + E_{FL}') \bullet R_{MS} = 1$.**

S No	Instruction	Action	Active Microprogram BITs	OC
1	Fetch OC	[IR] ← [PM]	S_{PC}, RD, L_{IR}, R_{MS}	-
2	JUD xx	[PC] ← xx	S_{PC}, RD, L_{PC}, R_{MS}	01
3	CUD xx	[SP] ← [SP]−1, [[SP]]← [PC], [PC] ← xx	E_{PC}, D_{SP}, S_{SP}, WR S_{PC}, RD, L_{PC}, R_{MS}	02
4	JUP xx	[PC] ← [PC] + xx	S_{PC}, RD, L_{PC}, I_{PC}, R_{MS}	03
5	CUP xx	[SP] ← [SP]−1, [[SP]]← [PC], [PC] ← [PC] + xx	E_{PC}, D_{SP}, S_{SP}, WR S_{PC}, RD, L_{PC}, I_{PC}, R_{MS}	04
6	JUA	[PC] ← [R0]	E_{R0}, L_{PC}, R_{MS}	05
7	CUA	[SP] ← [SP]−1,[[SP]]← [PC], [PC] ← [R0]	E_{PC}, D_{SP}, S_{SP}, WR E_{R0}, L_{PC}, R_{MS}	06
8	RTU	[PC]←[[SP]], [SP]←[SP]+1	I_{SP}, S_{SP}, RD, L_{PC}, R_{MS}	07
9	JCA <fl>	[PC] ← [R0] if <fl> = 1	E_{FL}, R_{MS} E_{R0}, L_{PC}, R_{MS}	08-0F
10	RTC <fl>	[PC]←[[SP]], [SP]←[SP]+1 if <fl> = 1	E_{FL}, R_{MS} I_{SP}, S_{SP}, RD, L_{PC}, R_{MS}	10-17
11	JCD <fl> xx	[PC] ← xx if <fl> = 1	S_{PC}, RD, L_{OR}, I_{PC}, E_{FL}, R_{MS} E_{OR}, L_{PC}, R_{MS}	18-1F
12	CCD <fl> xx	[SP] ← [SP]−1, [[SP]]← [PC], [PC] ← xx if <fl> = 1	S_{PC}, RD, L_{OR}, I_{PC}, E_{FL}, R_{MS} E_{PC}, D_{SP}, S_{SP}, WR E_{OR}, L_{PC}, R_{MS}	20-27
13	JCP <fl> xx	[PC] ← [PC] + xx if <fl> = 1	S_{PC}, RD, L_{OR}, I_{PC}, E_{FL}, R_{MS} E_{OR}, L_{PC}, I_{PC}, R_{MS}	28-2F
14	CCP <fl> xx	[SP] ← [SP]−1, [[SP]]← [PC], [PC] ← [PC] + xx if <fl> = 1	S_{PC}, RD, L_{OR}, I_{PC}, E_{FL}, R_{MS} E_{PC}, D_{SP}, S_{SP}, WR E_{OR}, L_{PC}, I_{PC}, R_{MS}	30-37
15	MVR <rn> xx	[<rn>] ← xx	S_{PC}, RD, L_{RN}, I_{PC}, R_{MS}	38-3F
16	NOT <rn>	[<rn>] ← [<rn>]'	E_{RN}, S_{AF}(A'), S_{AL}, L_{RN}, R_{MS}	40-47
17	CLR	<rn> ← 0, n = 0...7	S_{AF}(0), L_{R0}, L_{RN}	48
18	LDA <rn>*	[<rn>] ← [[R0]]	RD, L_{RN}, R_{MS}	49-4F
19	INC <rn>	[<rn>] ← [<rn>] + 1	E_{RN}, S_{AF}(A+1), S_{AL}, L_{RN}, R_{MS}	50-57
20	STS	[[R0]] ← [SP]	E_{SP}, WR, R_{MS}	58
21	STA <rn>*	[[R0]] ← [<rn>]	E_{RN}, WR, R_{MS}	59-5F
22	DCR <rn>	[<rn>] ← [<rn>] – 1	E_{RN}, S_{AF}(A–1), S_{AL}, L_{RN}, R_{MS}	60-67
23	MVP xx	[PC] ← xx	S_{PC}, RD, L_{PC}, R_{MS}	68
24	POP <rn>	[<rn>] ← [[SP]], [SP] ← [SP]+1	S_{SP}, RD, L_{RN}, I_{SP}, R_{MS}	69-6F

25	MVS xx	[SP] ← xx	S_{PC}, RD, L_{SP}, I_{PC}, R_{MS}	70
26	RLA <n>*	Rotate [R0] Left by n Bits, CY ← B_{8-n}	E_{R0}, $S_{AF}((2^n)*A)$, S_{AL}, L_{R0}, R_{MS}	71-77
27	NOP	No action	R_{MS}	78
28	PSH <rn>	[SP] ← [SP]–1, [[SP]] ← [<rn>]	D_{SP}, S_{SP}, E_{RN}, WR, R_{MS}	79-7F
29	ADA <rn>	[R0] ← [R0] + [<rn>]	E_{RN}, L_{OR} E_{R0}, S_{AF}(A+B), L_{R0}, R_{MS}	80-87
30	ADI <rn> xx	[<rn>] ← [<rn>] + xx	S_{PC}, RD, L_{OR}, I_{PC} E_{RN}, S_{AF}(A+B), L_{RN}, R_{MS}	88-8F
31	SBA <rn>	[R0] ← [R0] – [<rn>]	E_{RN}, L_{OR} E_{R0}, S_{AF}(A–B), L_{R0}, R_{MS}	90-97
32	SBI <rn> xx	[<rn>] ← [<rn>] – xx	S_{PC}, RD, L_{OR}, I_{PC} E_{RN}, S_{AF}(A–B), L_{RN}, R_{MS}	98-9F
33	ACA <rn>	[R0] ← [R0] + [<rn>] + C	E_{RN}, L_{OR} E_{R0}, S_{AF}(A+B+C), L_{R0}, R_{MS}	A0-A7
34	ACI <rn> xx	[<rn>] ← [<rn>] + xx + C	S_{PC}, RD, L_{OR}, I_{PC} E_{RN}, S_{AF}(A+B+C), L_{RN}, R_{MS}	A8-AF
35	SCA <rn>	[R0] ← [R0] – [<rn>] – C	E_{RN}, L_{OR} E_{R0}, S_{AF}(A–B-C), L_{R0}, R_{MS}	B0-B7
36	SCI <rn> xx	[<rn>] ← [<rn>] – xx – C	S_{PC}, RD, L_{OR}, I_{PC} E_{RN}, S_{AF}(A–B–C), L_{RN}, R_{MS}	B8-BF
37	ANA <rn>	[R0] ← [R0] ∧ [<rn>]	E_{RN}, L_{OR} E_{R0}, S_{AF}(A∧B), L_{R0}, R_{MS}	C0-C7
38	ANI <rn> xx	[<rn>] ← [<rn>] ∧ xx	S_{PC}, RD, L_{OR}, I_{PC} E_{RN}, S_{AF}(A∧B), L_{RN}, R_{MS}	C8-CF
39	ORA <rn>	[R0] ← [R0] ∨ [<rn>]	E_{RN}, L_{OR} E_{R0}, S_{AF}(A∨B), L_{R0}, R_{MS}	D0-D7
40	ORI <rn> xx	[<rn>] ← [<rn>] ∨ xx	S_{PC}, RD, L_{OR}, I_{PC} E_{RN}, S_{AF}(A∨B), L_{RN}, R_{MS}	D8-DF
41	XRA <rn>	[R0] ← [R0] ⊕ [<rn>]	E_{RN}, L_{OR} E_{R0}, S_{AF}(A⊕B), L_{R0}, R_{MS}	E0-E7
42	XRI <rn> xx	[<rn>] ← [<rn>] ⊕ xx	S_{PC}, RD, L_{OR}, I_{PC} E_{RN}, S_{AF}(A⊕B), L_{RN}, R_{MS}	E8-EF
43	XNA <rn>	[R0] ← [R0] ⊕ [<rn>]	E_{RN}, L_{OR} E_{R0}, S_{AF}(A⊕B′), L_{R0}, R_{MS}	F0-F7
44	XNI <rn> xx	[<rn>] ← [<rn>] ⊕ xx	S_{PC}, RD, L_{OR}, I_{PC} E_{RN}, S_{AF}(A⊕B′), L_{RN}, R_{MS}	F8-FF