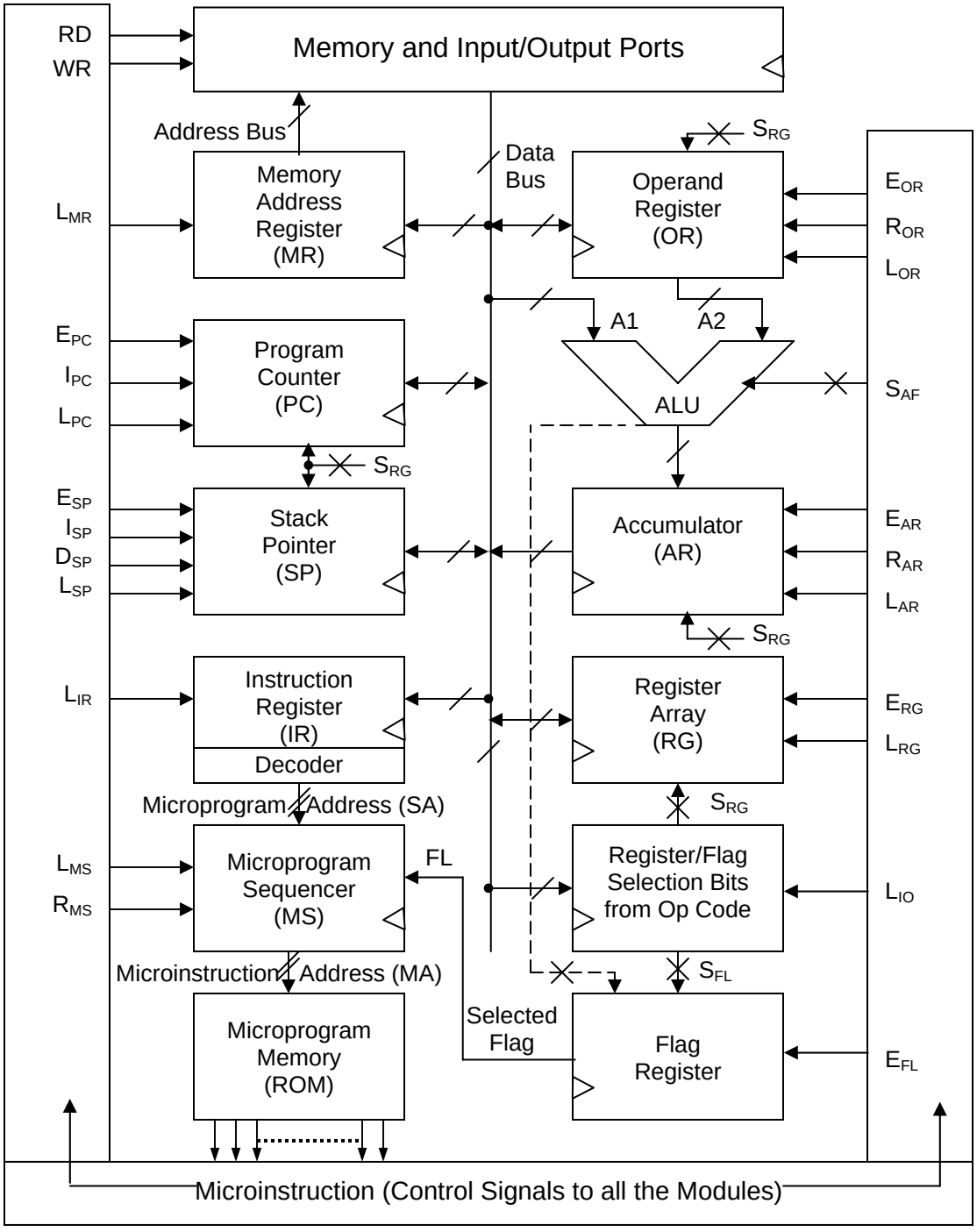
Instruction Set for the Single-Bus Processor Architecture			
Sl.No.	**Instruction**	**Action**	**Op Code (Hex)**
1	nop	No action	00
2	stop	Exit the program	01
3	adi xx	[AR] ← [AR] + xx	02
4	subi xx	[AR] ← [AR] – xx	03
5	xri xx	[AR] ← [AR] ⊕ xx	04
6	ani xx	[AR] ← [AR] ∧ xx	05
7	ori xx	[AR] ← [AR] ∨ xx	06
8	cmi xx	[AR] – xx (Flags only)	07
9-16	ret<FL>	[PC] ← [[SP]], [SP] ← [SP]+1 if <FL> = 1	08 to 0F
17	add <R>	[AR] ← [AR] + [<R>]	10-1F (n:0-F)
18	sub <R>	[AR] ← [AR] – [<R>]	20-2F (n:0-F)
19	xor <R>	[AR] ← [AR] ⊕ [<R>]	30-3F(n:0-F)
20	and <R>	[AR] ← [AR] ∧ [<R>]	40-4F(n:0-F)
21	or <R>	[AR] ← [AR] ∨ [<R>]	50-5F(n:0-F)
22	cmp <R>	[AR] – [<R>] (Flags only)	60-6F(n:0-F)
23	movs <R>	[OR] ← [<R>], [AR] ← [<R>]	70-7F(n:0-F)
24	movd <R>	[<R>] ← [AR]	80-8F(n:0-F)
25	movi <R> xx	[<R>] ← xx	90-9F(n:0-F)
26	store <R>	[[AR]] ← [<R>]	A0-AF(n:0-F)
27	load <R>	[<R>] ← [[AR]]	B0-BF(n:0-F)
28	push <R>	[SP] ← [SP]–1, [[SP]] ← [<R>]	C0-CF(n:0-F)
29	pop <R>	[<R>] ← [[SP]], [SP] ← [SP]+1	D0-DF(n:0-F)
30-37	jmpd<FL> xx	[PC] ← xx if <FL> = 1	E0 to E7
38-45	jmpr<FL>	[PC] ← [AR] if <FL> = 1	E8 to EF
46-53	cd<FL> xx	[SP] ← [SP]–1, [[SP]]← [PC], [PC] ← xx if <FL> = 1	F0 to F7
54-61	cr<FL>	[SP] ← [SP]–1, [[SP]]← PC], [PC] ← [AR] if <FL> = 1	F8 to FF
Flags: Zero (Z), Carry (CY), Sign (S), Parity (P) <FL> = u/z/nz/c/nc/p/m/op ⇒ FL = 0/Z/Z’/CY/CY’/S/S’/1 if Parity odd <R> = r1/r2/r3/r4/r5/r6/r7/r8/r9/r10/r11/PC/SP/AR/OR			

ALU Function Codes:

0000	0001	0010	0011	0100	0101	0110	1111
-	ADD	SUB	XOR	AND	OR	CMP	A1

Architecture based on a Single Internal Data Bus



Nomenclature:

E_{XY} : Enables the Output of the module XY on to the Data Bus;
L_{XY} : Enables the Data Input applied to module XY to be loaded into the module at the next Active Clock edge;
$I_{PC,}$ I_{SP}, D_{SP} : Enable Incrementing PC, Incrementing SP and Decrementing SP at the next Active Clock edge;
S_{XY} : Select bits for module XY; R_{0R} / R_{AR} : Clears the register OR / AR

The actual CLEAR input applied to the Microprogram Sequencer is given by:

$CLEAR_{MS} = R_{MS} \bullet FL' \bullet E_{FL}$, where FL is the selected FLAG and E_{FL} is the Flag Register output enable control.

Hence in conditional Branch instructions, the MS is cleared resulting in the next FETCH, if $R_{MS} = 1$ AND the selected FLAG is not set (FL = 0); for unconditional Branch, FL = 0, and hence the MS is cleared if $R_{MS} = 1$.