Microinstruction Format

A	amux	B	bmux	C	cmux	ALU	u	rd	wr	cond	addr	
5	1	5	1	5	1	4	1	1	1	4	11	width

Field

`A` Specifies register that inputs to the A multiplexer

`amux` Controls `A` multiplexer:
`amux` = 0 then `A` field in `mir` drives A decoder
`amux` = 1 then `A` field in `ir` drives A decoder

`B` Specifies register that inputs to the B multiplexer

`bmux` Controls B multiplexer:
`bmux` = 0 then `B` field in `mir` drives B decoder
`bmux` = 1 then `B` field in `ir` drives B decoder

`C` Specifies register that inputs to the C multiplexer

`cmux` Controls C multiplexer:
`cmux` = 0 then `C` field in `mir` drives C decoder
`cmux` = 1 then `C` field in `ir` drives C decoder

`alu` Specifies ALU operation:

F_3	F_2	F_1	F_0		Mnemonic	Output	Flags Set
0	0	0	0	(0)	`nop`	left	
0	0	0	1	(1)	`not`	~left	`nz`
0	0	1	0	(2)	`and`	left & right	`nz`
0	0	1	1	(3)	`sext`	left sign ext, (rt = mask)	`nz`
0	1	0	0	(4)	`add`	left + right	`nzcv`
0	1	0	1	(5)	`sub`	left – right	`nzcv`
0	1	1	0	(6)	`mul`	left * right	`nz`
0	1	1	1	(7)	`div`	left / right	`nz`
1	0	0	0	(8)	`rem`	left % right	`nz`
1	0	0	1	(9)	`or`	left | right	`nz`
1	0	1	0	(10)	`xor`	left ^ right	`nz`
1	0	1	1	(11)	`sll`	left << right (logical)	`nzc`
1	1	0	0	(12)	`srl`	left >> right (logical	`nzc`
1	1	0	1	(13)	`sra`	left >> right (arithmetic)	`nzc`
1	1	1	0	(14)	`rol`	left << right (rotate)	`nzc`
1	1	1	1	(15)	`ror`	left >> right (rotate)	`nzc`

`u` User `n`, `z`, `c`, `v` flags used instead of system flags

`rd` Initiates memory read from address in `mar`
`wr` Initiates memory write of data in `mdr` to address in `mar`

`cond` Specifies branch condition:

	Mnemonic	Branch if	Branch on
0	`nobr`		never
1	`zer`	`z` = 1	zero or equal
2	`!zer`	`z` = 0	not zero or not equal
3	`neg`	`n` = 1	negative
4	`!neg`	`n` = 0	not negative
5	`cy` or `<`	`c` = 1	less than (unsigned cmp)
6	`!cy` or `>=`	`c` = 0	grt or eq (unsigned cmp/overflow)
7	`v`	`v` = 1	signed overflow
8	`pos`	`n` = `z`	positive
9	`lt`	`n` != `v`	less than (signed cmp)
10	`le`	`n` != `v` or `z` = 1	less than or equal (signed cmp)
11	`gt`	`n` = `v` and `z` = 0	greater (signed cmp)
12	`ge`	`n` = `v`	greater than or equal (signed cmp)
13	`<=`	`c` = 1 or `z` = 1	less than or equal (unsigned cmp)
14	`>`	`c` = 0 and `z` = 0	greater than (unsigned cmp)
15	`br`		always

`addr` branch-to address

In the enhanced LCC (see Chapter 10 in “Constructing a Microprogrammed Computer Second Edition”),

- each microinstruction has a `mmux` bit in the rightmost position that controls the `mpc` multiplexer
- the `mpc` is 12 bits wide
- the `addr` field in each microinstruction is 12 bits wide
- microstore can hold up to 4096 microinstructions.

Register Names

Reg Num	Name	Initial Contents	Function
0	r0 or ac		accumulator register
1	r1		
2	r2		
3	r3		
4	r4		
5	r5 or fp		frame pointer register
6	r6 or sp		stack pointer register
7	r7 or lr		link register
8	r8 or 3	0x0003	
9	r9 or 4	0x0004	
10	r10 or 5	0x0005	
11	r11 or omask	0xf000	opcode mask
12	r12 or cmask	0x01e0	count mask (for shifts)
13	r13 or bit5	0x0020	
14	r14 or bit11	0x0800	
15	r15 or bit15	0x8000	
16	r16 or m3	0x0007	
17	r17 or m4	0x000f	
18	r18 or m5	0x001f	
19	r19 or m6	0x003f	
20	r20 or m8	0x00ff	
21	r21 or m9	0x01ff	
22	r22 or m11	0x07ff	
23	r23 or m12	0x0fff	
24	r24 or ir		machine instruction register
25	r25 or pc		program counter register
26	r26 or temp		
27	r27 or dc		decoding register
28	r28 or 1	0x0001	constant 1
29	r29 or mar		memory address register
30	r30 or mdr		memory data register
31	r31 or 0	0x0000	constant 0 (read-only register)