CG2028 COMPUTER ORGANIZATION

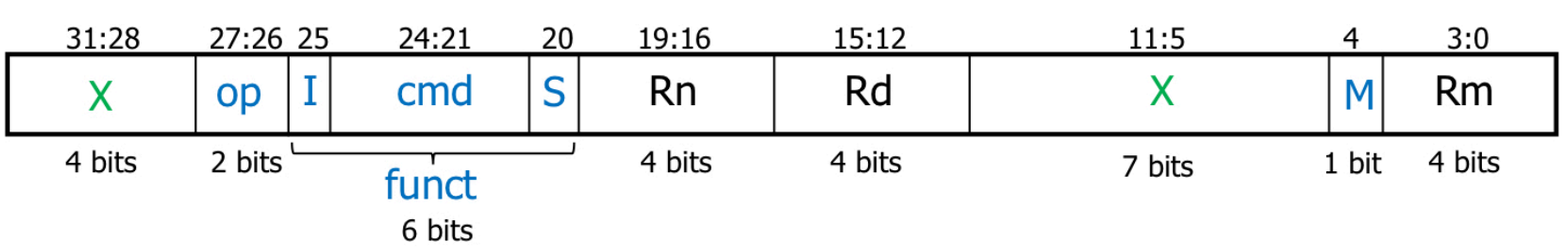
TUTORIAL 4: SINGLE CYCLE PROCESSOR DESIGN

GU JING

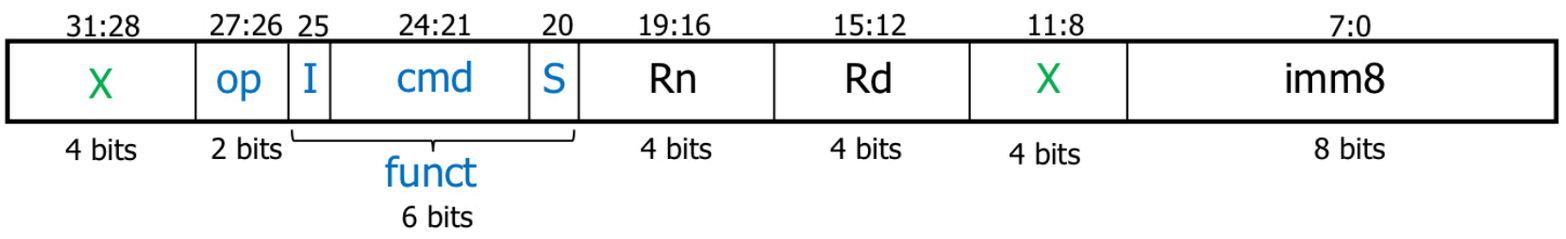
Q1. Write the assembly language instructions (consider the extended formats given in slides 37-40) corresponding to the machine codes below.

a. 0x0224201C b. 0x0024201C c. 0x0404001C d. 0x0804001C

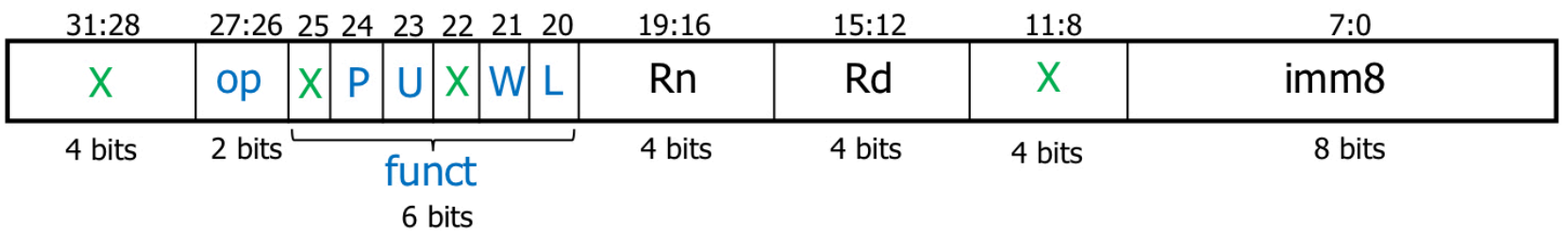
OP=00



OP=00

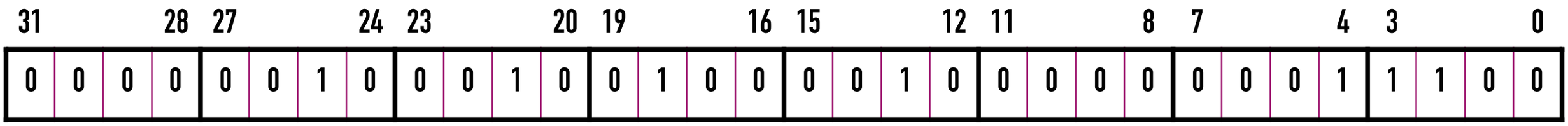


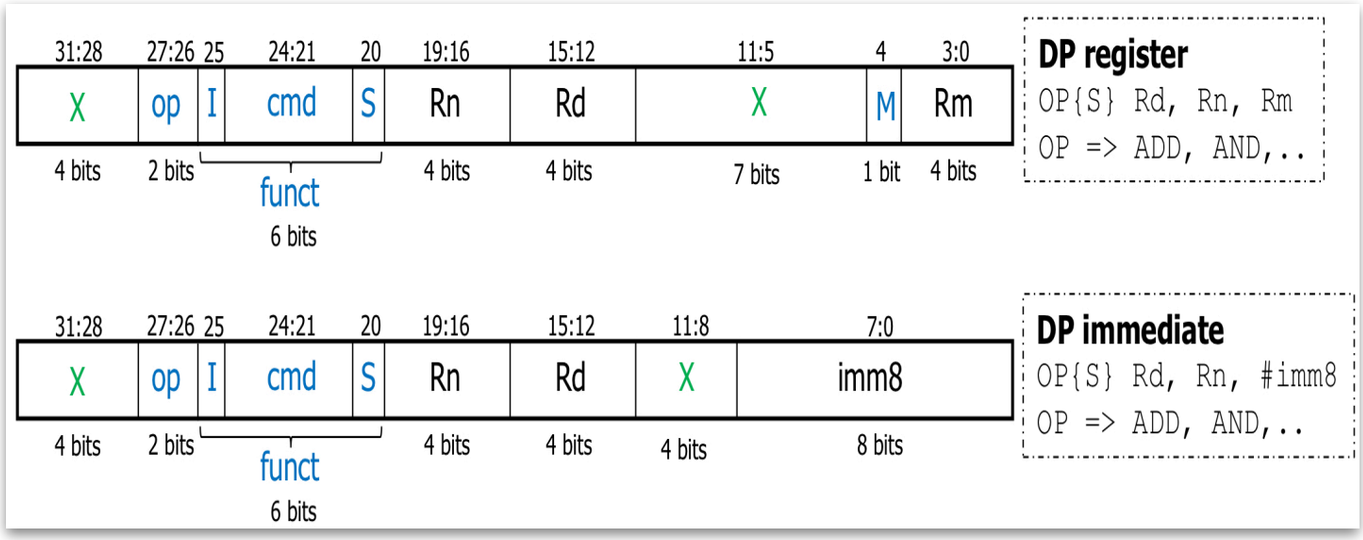
OP=01



OP=10

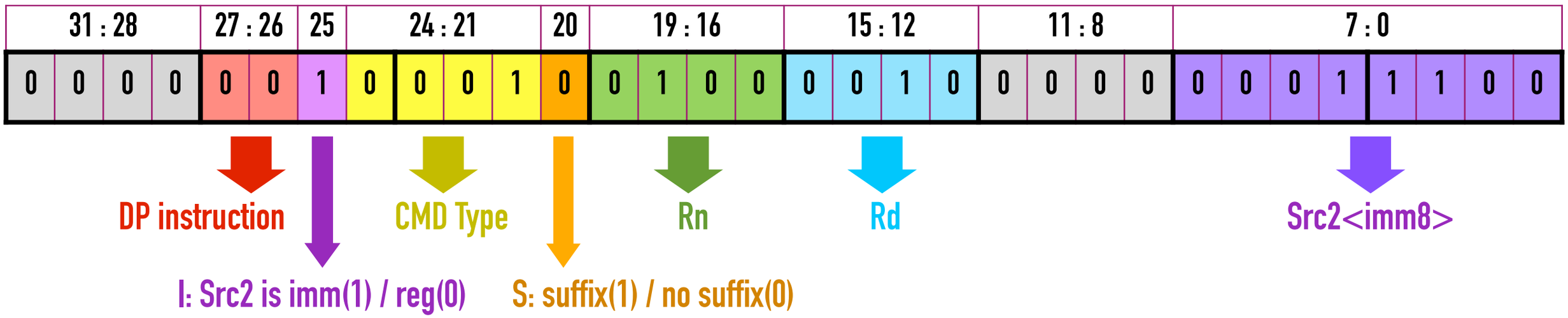
Q1 a. 0x0224201C

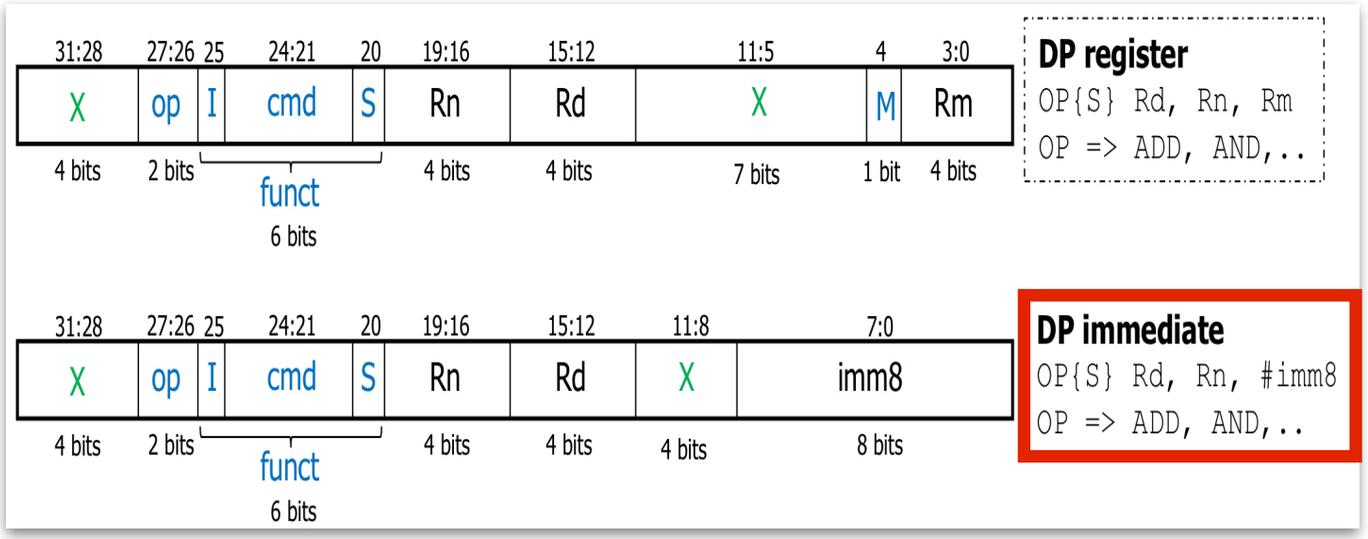




cmd	Instruction	Operation
0000	AND	Logical AND
0001	EOR	Logical Exclusive OR
0010	SUB	Subtract
0011	RSB	Reverse Subtract
0100	ADD	Add
0101	ADC	Add with Carry
0110	SBC	Subtract with Carry
0111	RSC	Reverse Subtract with Carry
1000	TST	Test Update flags after AND
1001	TEQ	Test Equivalence Update flags after EOR
1010	CMP	Compare Update flags after SUB
1011	CMN	Compare Negated Update flags after ADD
1100	ORR	Logical OR
1101	MOV	Move
1110	BIC	Bit Clear
1111	MVN	Move Not

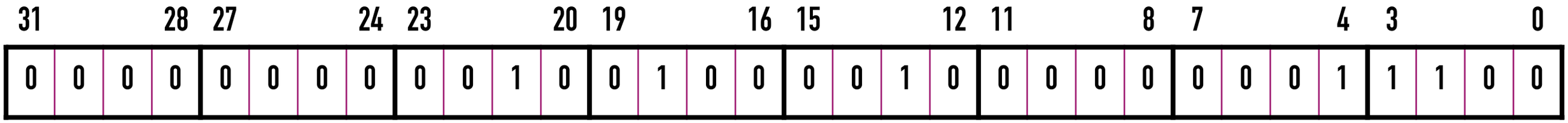
Q1 a. 0x0224201C: EOR R2, R4, #0x1C

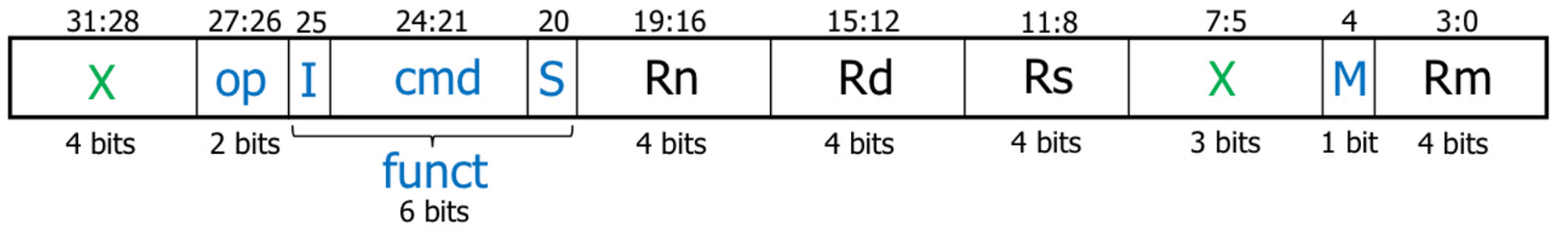




cmd	Instruction	Operation
0000	AND	Logical AND
0001	EOR	Logical Exclusive OR
0010	SUB	Subtract
0011	RSB	Reverse Subtract
0100	ADD	Add
0101	ADC	Add with Carry
0110	SBC	Subtract with Carry
0111	RSC	Reverse Subtract with Carry
1000	TST	Test Update flags after AND
1001	TEQ	Test Equivalence Update flags after EOR
1010	CMP	Compare Update flags after SUB
1011	CMN	Compare Negated Update flags after ADD
1100	ORR	Logical OR
1101	MOV	Move
1110	BIC	Bit Clear
1111	MVN	Move Not

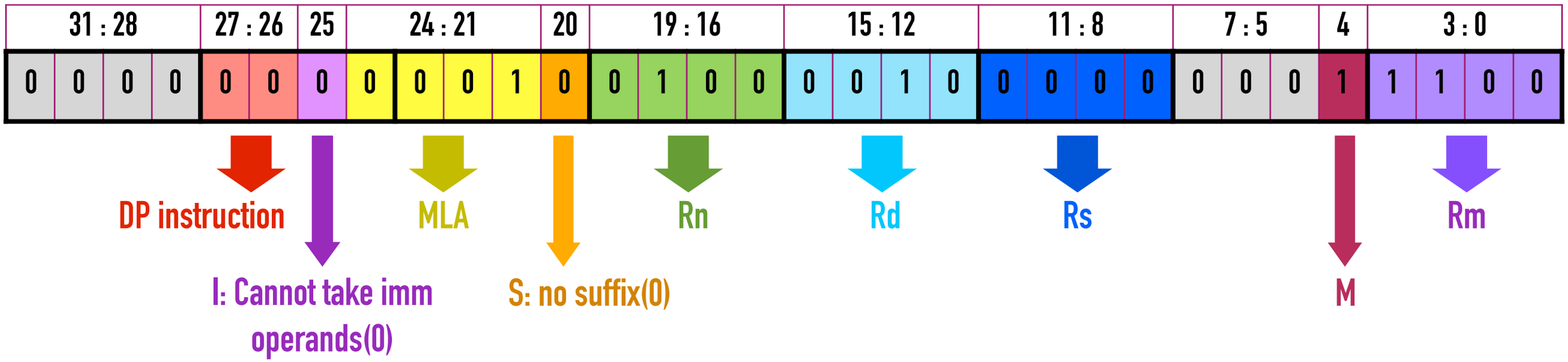
Q1 **b. 0x0024201C**

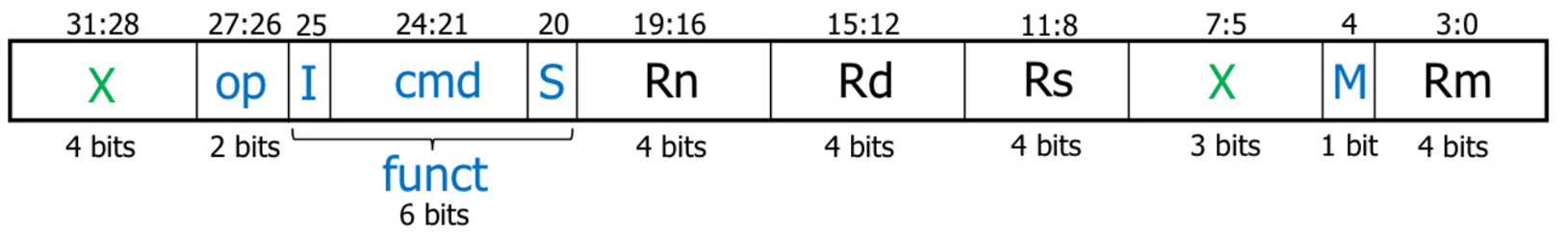




- `MUL Rd, Rm, Rs` (*Rd* = *Rm***Rs*)
- `MLA Rd, Rm, Rs, Rn` (*Rd* = *Rn* + *Rm***Rs*)
- cmd = 0b0000 for MUL, 0b0001 for MLA
- M = 0b0 -> usual DP instructions such as ADD, AND,..
  0b1 -> MUL and MLA

Q1 b. 0x0024201C: MLA R2, R12, R0, R4

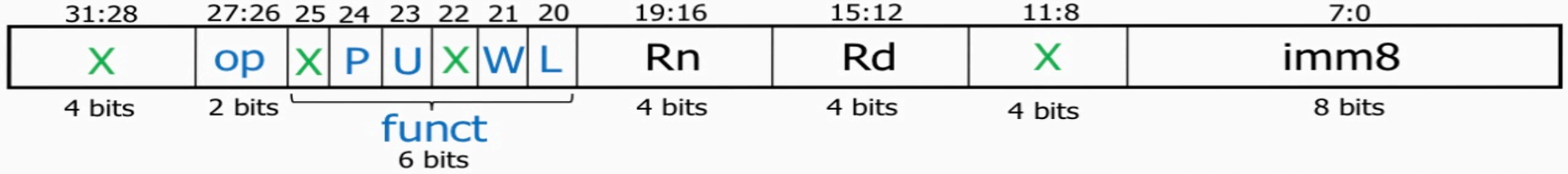




- `MUL Rd, Rm, Rs` (*Rd = Rm*Rs*)
- `MLA Rd, Rm, Rs, Rn` (*Rd = Rn + Rm*Rs*)
- cmd = 0b0000 for MUL, 0b0001 for MLA
- M = 0b0 -> usual DP instructions such as ADD, AND,..
  0b1 -> MUL and MLA

Q1 **c. 0x0404001C**

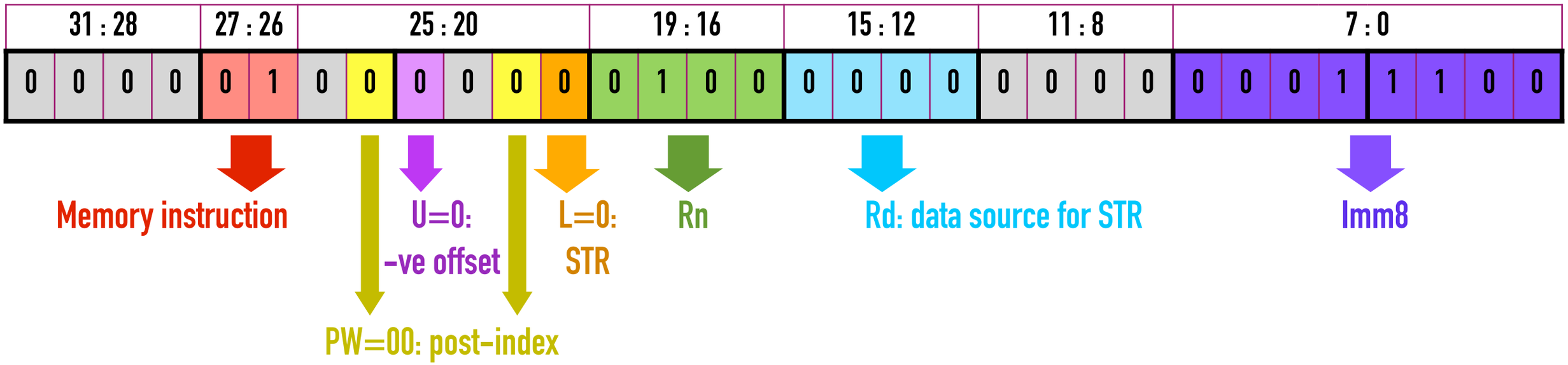
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0

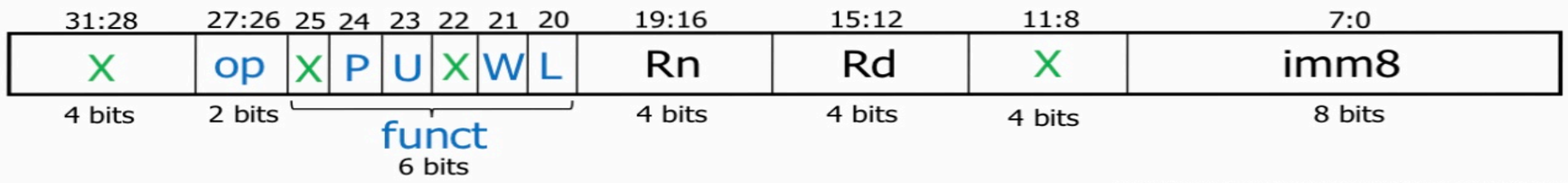



```
OP Rd, [Rn, #±imm8]
OP => LDR, STR
```

- Encodes LDR, STR
  - op : 0b01 for memory instructions
  - funct : 6 control bits
  - U : Add
    - 0b1 -> offset is positive, i.e., effective address = *Rn* + *imm8*
    - 0b0 -> the offset is negative, i.e., effective address = *Rn* – *imm8*
  - L = 0b1 for load; 0b0 for store
  - PW = 0b10
  - *Rn* : base register
  - *Rd* : destination (load), source (store)
  - *imm8* : magnitude of offset

Q1 c. 0x0404001C: STR R0, [R4], #-0x1C

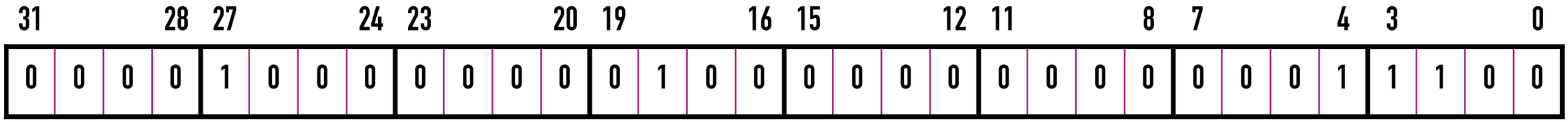


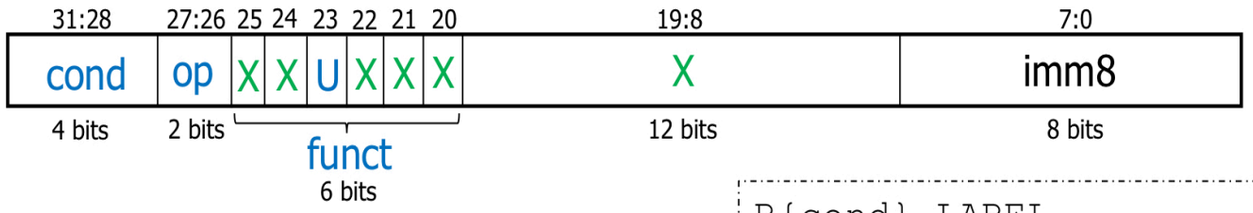



```
OP Rd, [Rn, #±imm8]
OP => LDR, STR
```

- Encodes LDR, STR
  - op : 0b01 for memory instructions
  - funct : 6 control bits
  - U : Add
    - 0b1 -> offset is positive, i.e., effective address = *Rn* + *imm8*
    - 0b0 -> the offset is negative, i.e., effective address = *Rn* − *imm8*
  - L = 0b1 for load; 0b0 for store
  - PW = 0b00 -> postindex 0b01 -> unsupported
    0b10 -> offset 0b11 -> preindex
  - *Rd* : destination (load), source (store)
  - *imm8* : magnitude of offset

Q1 d. 0x0804001C



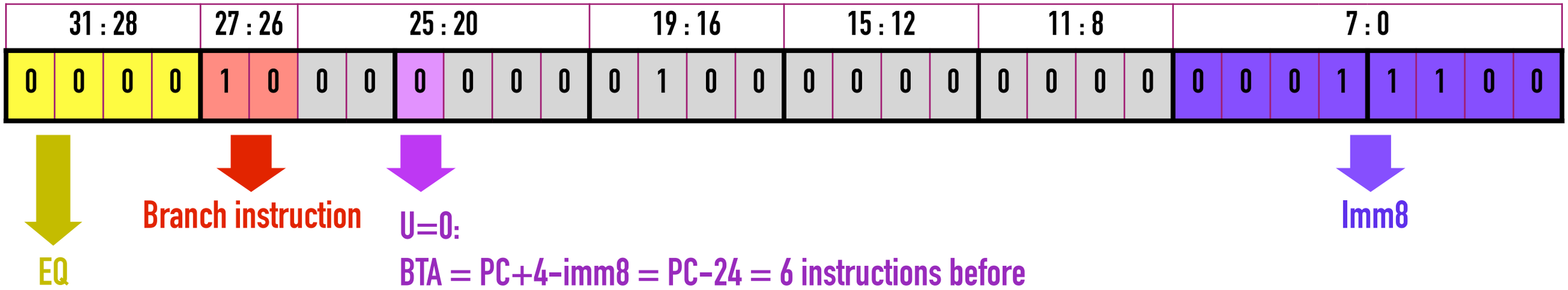


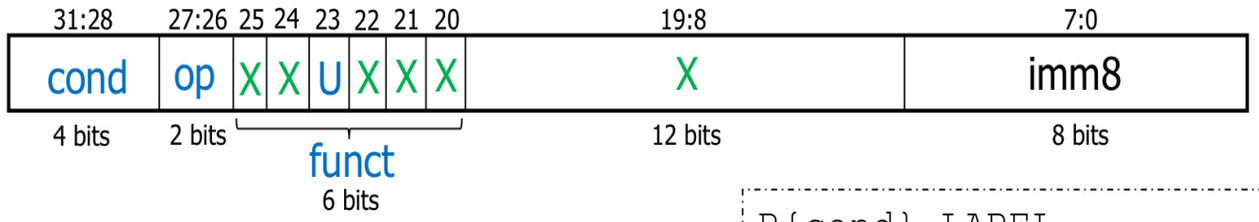

```
B{cond} LABEL
LABEL encoded as #±imm8
```

- Encodes B{cond}
  - cond : condition to be true for the branch to be taken
    - EQ = 0b0000
    - AL (always a.k.a unconditional) = 0b1110
  - op = 0b10 for branch instructions
  - *imm8* : 8-bit immediate encoding Branch Target Address (BTA)
    - BTA = address corresponding to LABEL = Next PC when branch taken
    - *imm8* = # of bytes BTA is away from current PC+4
  - U : add
    - 0b1 -> BTA = PC+4+*imm8*; 0b0 -> BTA = PC+4-*imm8*

cond	Mnemonic	Name	Condition Checked
0000	EQ	Equal	Z
0001	NE	Not equal	$\bar{Z}$
0010	CS / HS	Carry set / Unsigned higher or same	C
0011	CC / LO	Carry clear / Unsigned lower	$\bar{C}$
0100	MI	Minus / Negative	N
0101	PL	Plus / Positive of zero	$\bar{N}$
0110	VS	Overflow / Overflow set	V
0111	VC	No overflow / Overflow clear	$\bar{V}$
1000	HI	Unsigned higher	$\bar{Z}C$
1001	LS	Unsigned lower or same	$Z\ OR\ \bar{C}$
1010	GE	Signed greater than or equal	$\overline{N \oplus V}$
1011	LT	Signed less than	$N \oplus V$
1100	GT	Signed greater than	$\bar{Z}(\overline{N \oplus V})$
1101	LE	Signed less than or equal	$Z\ OR\ (N \oplus V)$
1110	AL (or none)	Always / unconditional	ignored

Q1 d. 0x0804001C: BEQ LABEL *Where LABEL is 6 instructions before BEQ*



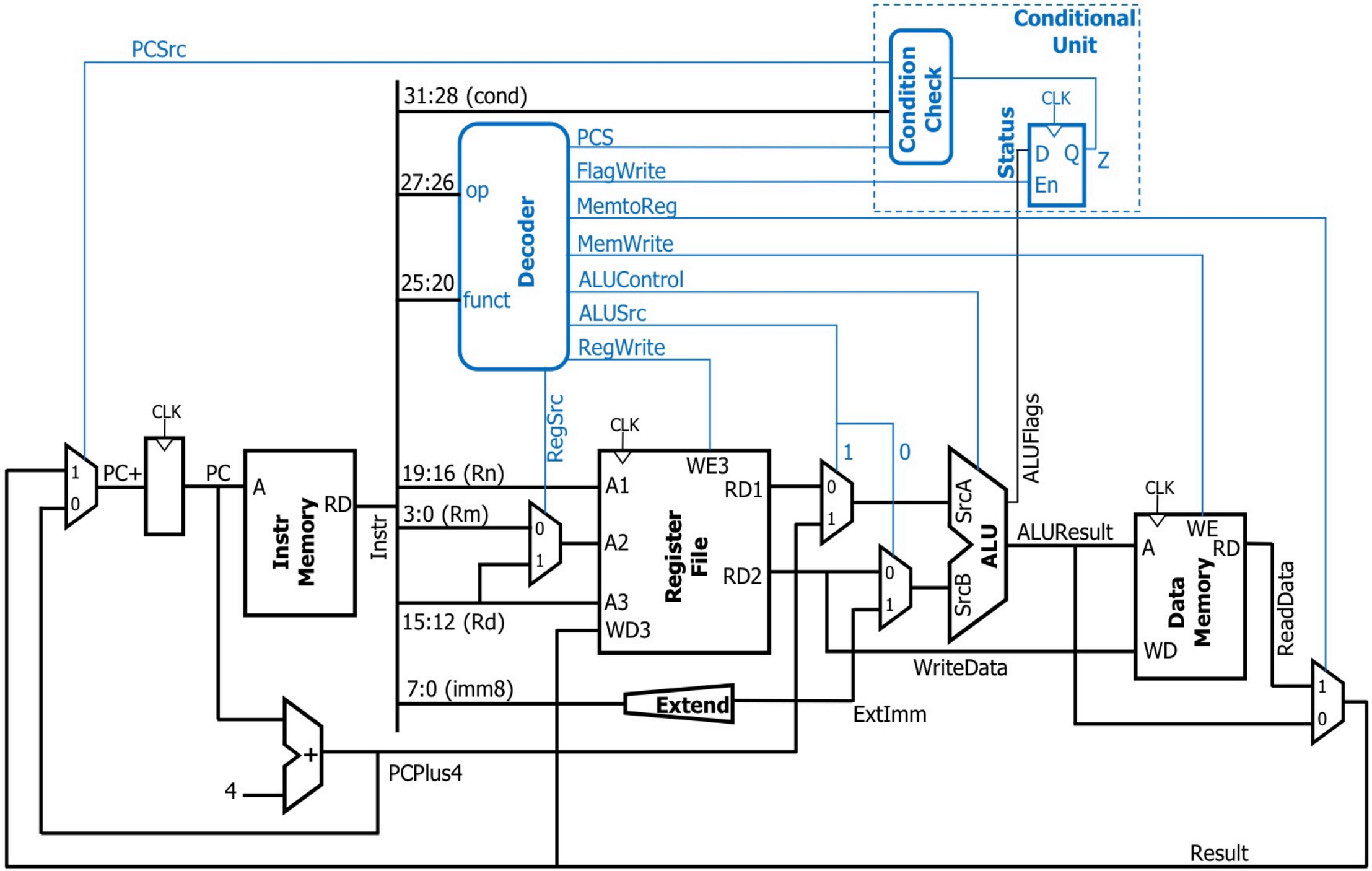



```
B{cond} LABEL
LABEL encoded as #±imm8
```

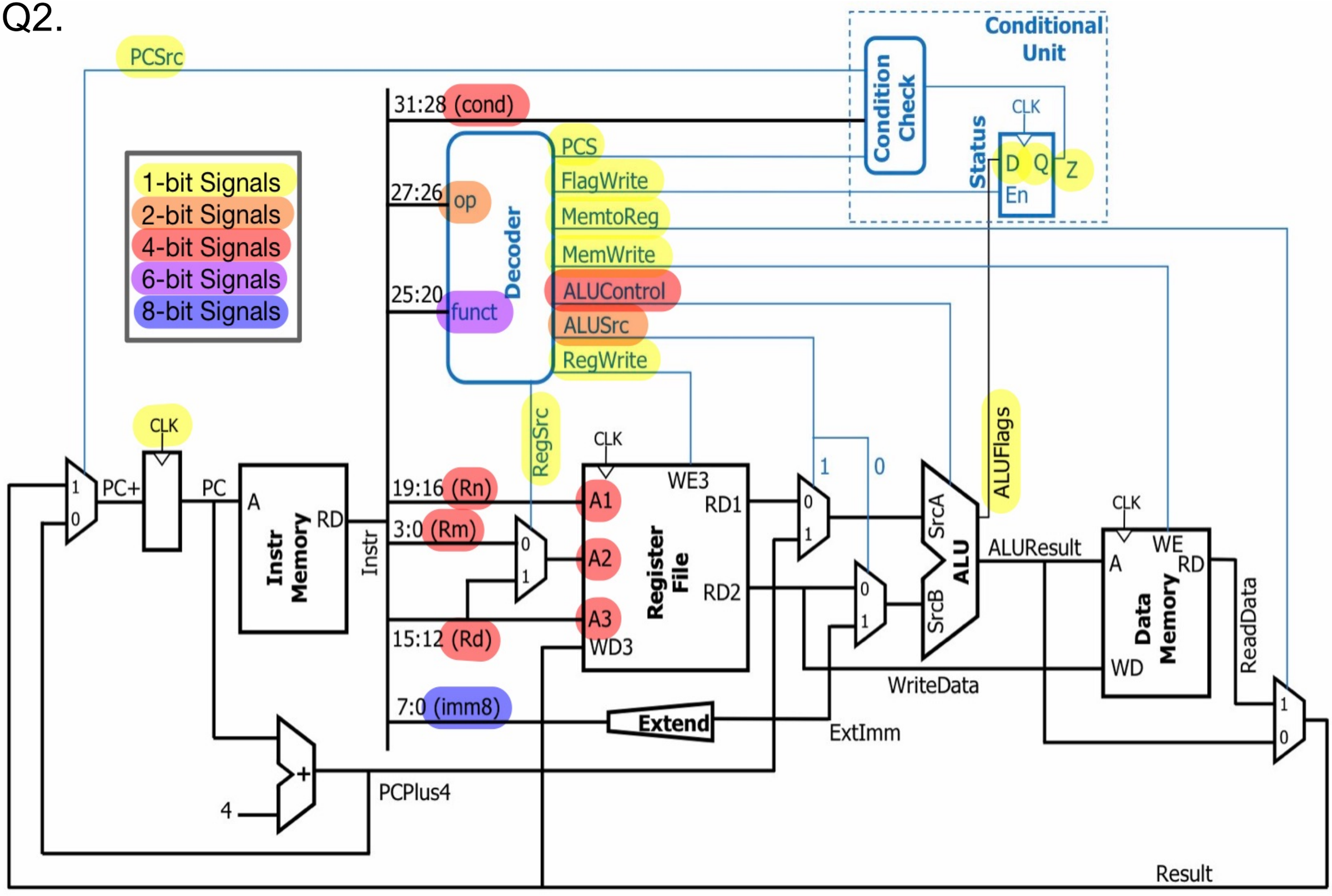
- Encodes B{cond}
  - cond : condition to be true for the branch to be taken
    - EQ = 0b0000
    - AL (always a.k.a unconditional) = 0b1110
  - op = 0b10 for branch instructions
  - *imm8* : 8-bit immediate encoding Branch Target Address (BTA)
    - BTA = address corresponding to LABEL = Next PC when branch taken
    - *imm8* = # of bytes BTA is away from current PC+4
  - U : add
    - 0b1 -> BTA = PC+4+*imm8*; 0b0 -> BTA = PC+4-*imm8*

cond	Mnemonic	Name	Condition Checked
0000	EQ	Equal	Z
0001	NE	Not equal	$\bar{Z}$
0010	CS / HS	Carry set / Unsigned higher or same	C
0011	CC / LO	Carry clear / Unsigned lower	$\bar{C}$
0100	MI	Minus / Negative	N
0101	PL	Plus / Positive of zero	$\bar{N}$
0110	VS	Overflow / Overflow set	V
0111	VC	No overflow / Overflow clear	$\bar{V}$
1000	HI	Unsigned higher	$\bar{Z}C$
1001	LS	Unsigned lower or same	$Z\ OR\ \bar{C}$
1010	GE	Signed greater than or equal	$\overline{N \oplus V}$
1011	LT	Signed less than	$N \oplus V$
1100	GT	Signed greater than	$\bar{Z}(\overline{N \oplus V})$
1101	LE	Signed less than or equal	$Z\ OR\ (N \oplus V)$
1110	AL (or none)	Always / unconditional	ignored

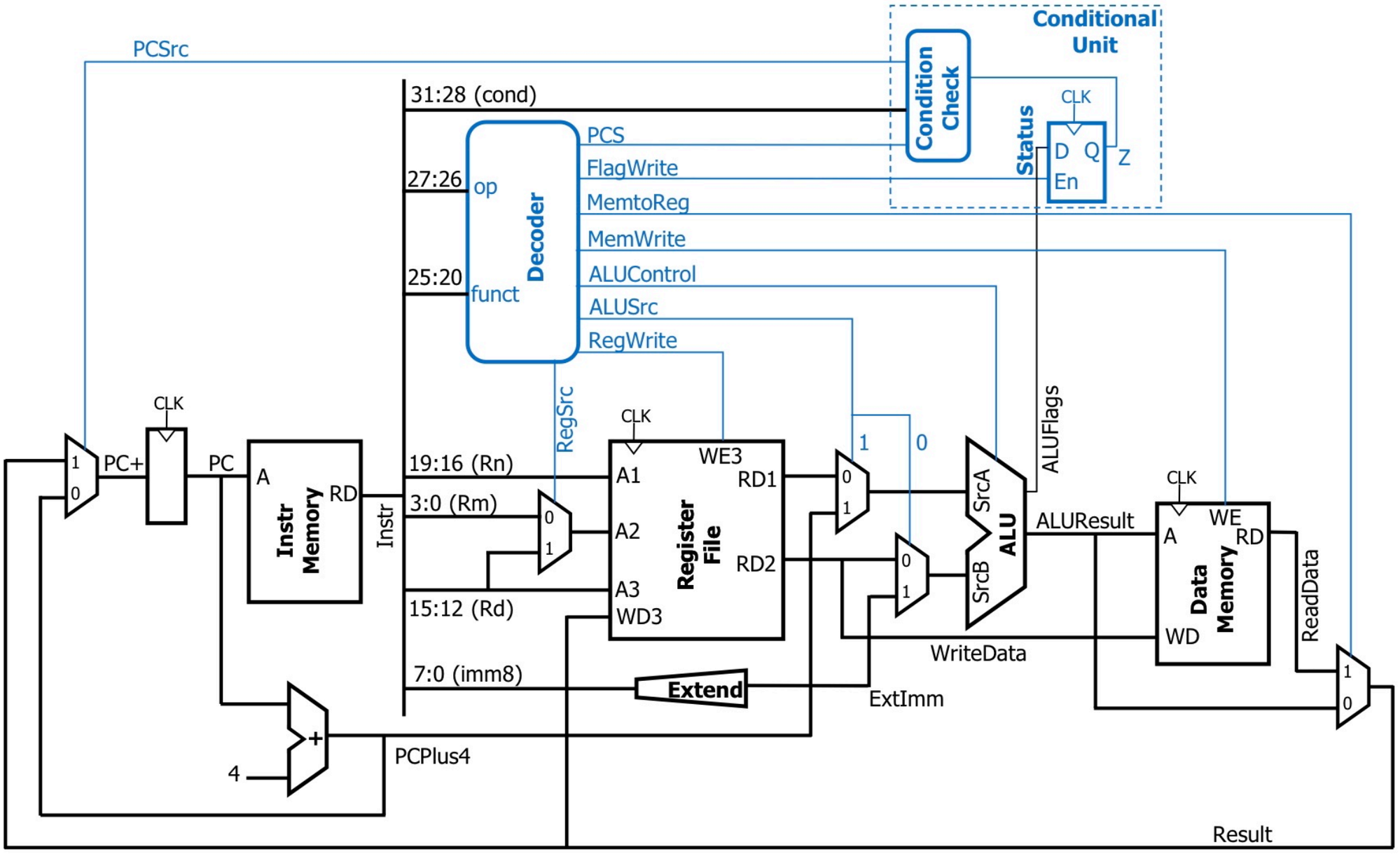
Q2. Annotate the bit widths for all the connections of the microarchitecture given in Slides 28 of Lecture 4.



Q2.


PCSrc
1-bit Signals
2-bit Signals
4-bit Signals
6-bit Signals
8-bit Signals
Conditional Unit
Condition Check
Status
CLK
D
Q
Z
En
31:28 (cond)
27:26
op
25:20
funct
Decoder
PCS
FlagWrite
MemtoReg
MemWrite
ALUControl
ALUSrc
RegWrite
RegSrc
ALUFlags
CLK
PC+
PC
A
Instr Memory
RD
Instr
19:16 (Rn)
3:0 (Rm)
15:12 (Rd)
7:0 (imm8)
A1
A2
A3
WD3
WE3
RD1
RD2
Register File
Extend
ExtImm
SrcA
SrcB
ALU
ALUResult
WriteData
WE
Data Memory
WD
ReadData
PCPlus4
4
Result

Q3. Modify the microarchitecture given in Lecture 4 to incorporate BNE instruction. Detail the datapath and control unit modifications required, including logic expressions for new control signals / existing control signals which need to be modified, if any.



CG2028 Tutorial 4: Single Cycle Processor Design

Q3.

31 : 28	27 : 26	25	24	23	22	21	20	19 : 8	7 : 0
Cond	Op	X	X	U	X	X	X	X	Imm8

2b10 for Branch instruction

U=1: BTA = PC+4+imm8
U=0: BTA = PC+4-imm8

4b0000: EQ
4b0001: NE
4b1110: Always

Integrate condition codes (NE/ EQ/...) with PCSrc:

BNE: PCSrc=1 if PCS=1 and Z=0

BEQ: PCSrc=1 if PCS=1 and Z=1