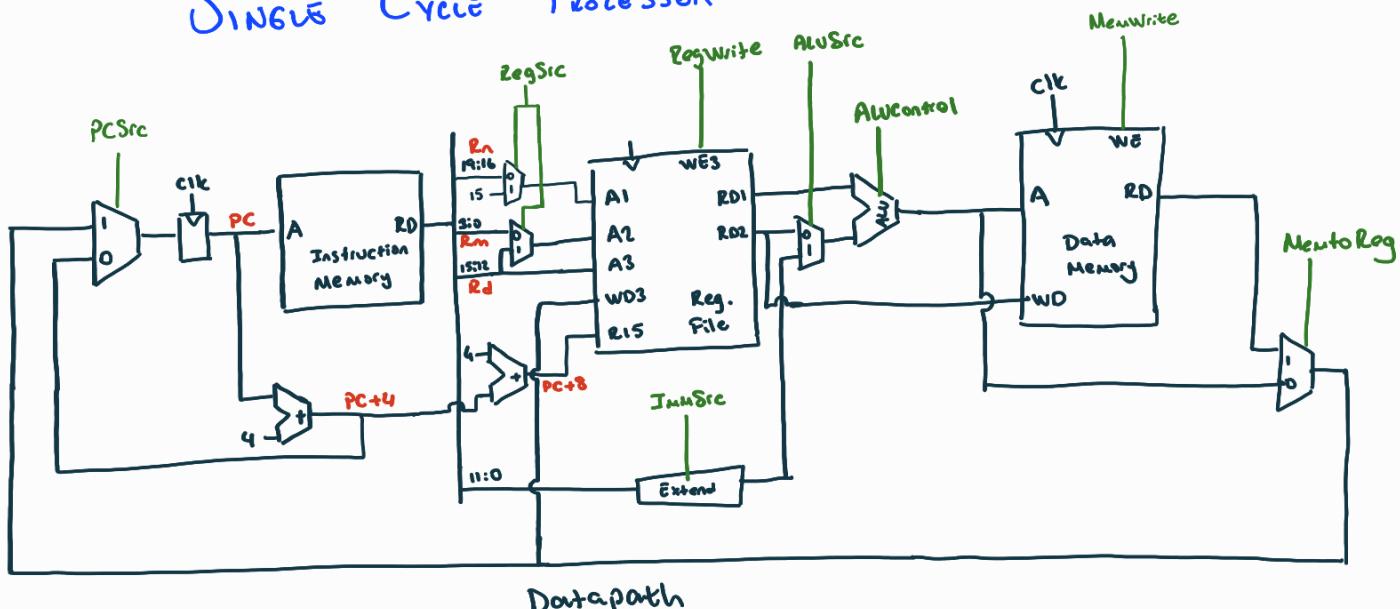
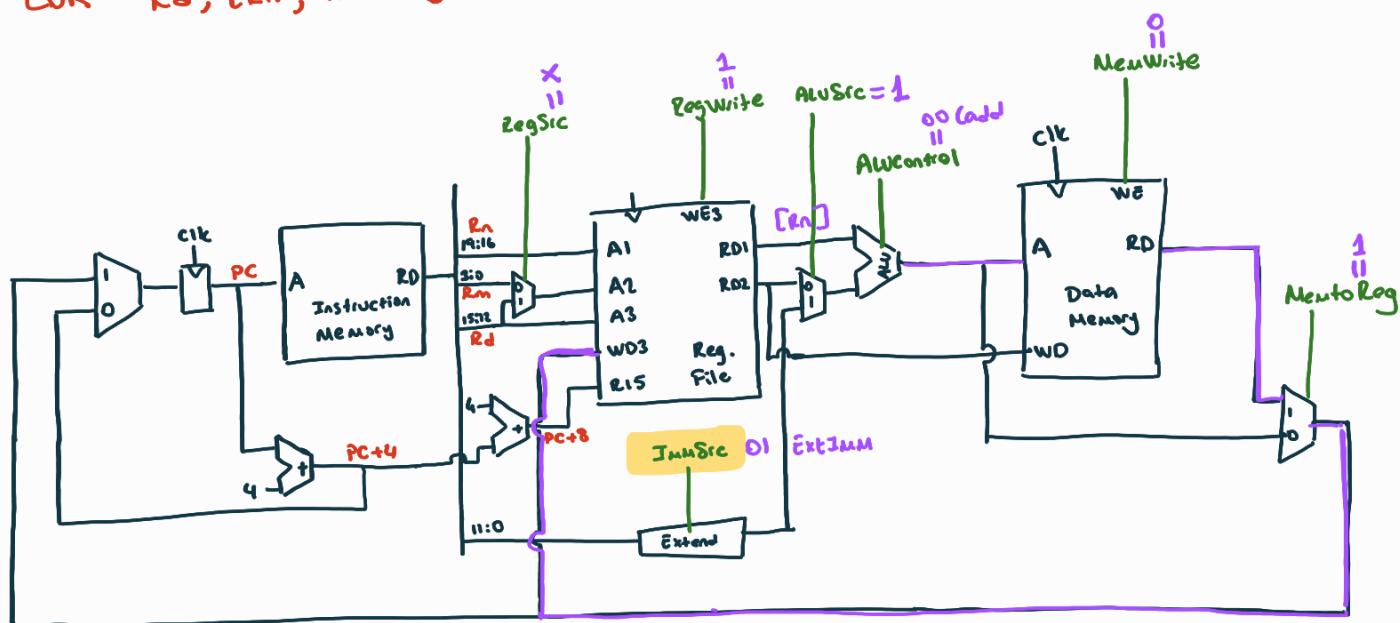


SINGLE CYCLE PROCESSOR

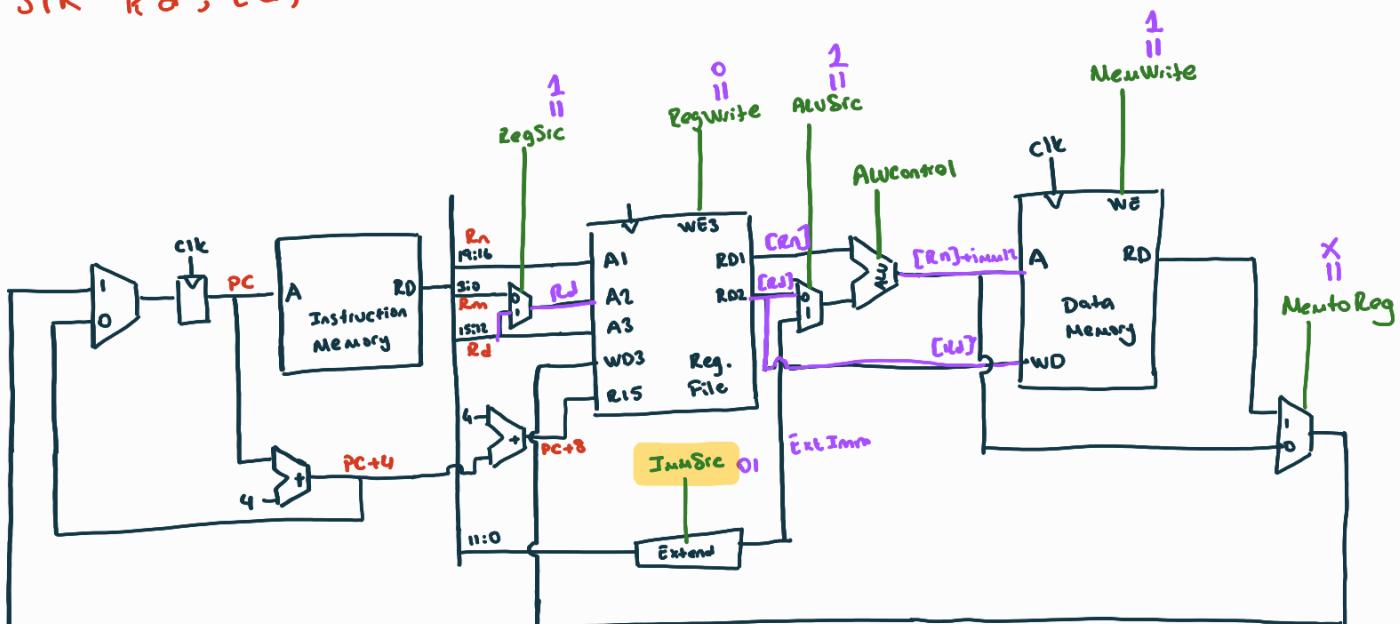


Datapath

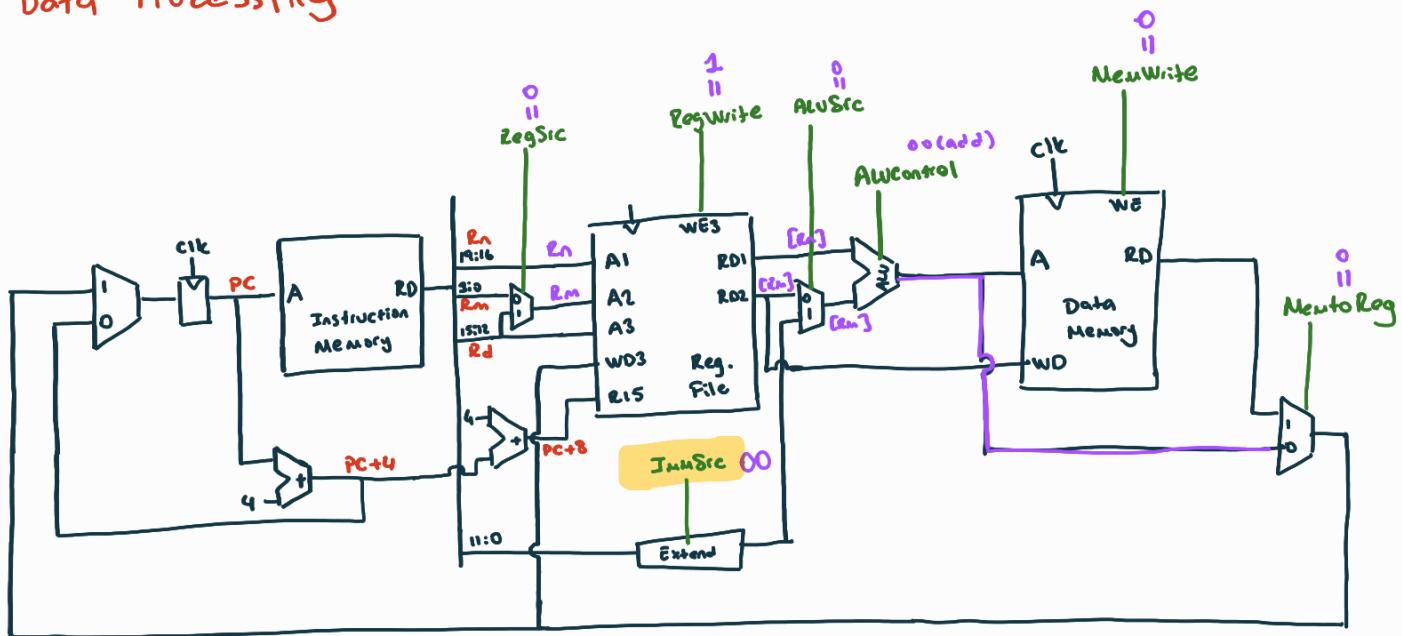
$$LDR \quad Rd, [Rn, imm12] \rightarrow RJ = M[Rn + imm12]$$



$$STR \quad Rd, [Rn, imm12] \rightarrow M[Rn + imm12] = Rd$$



Data Processing

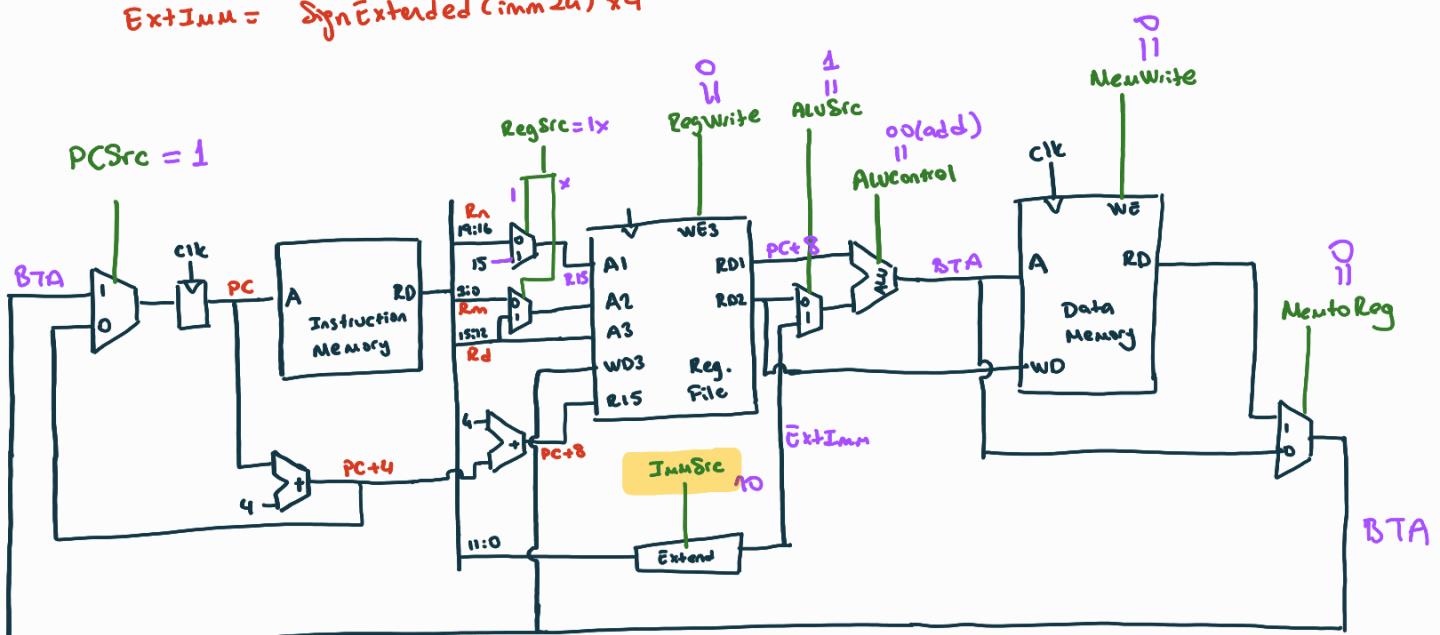


Branch Instruction

$$PC = BTA$$

$$BTA = (\bar{E}k + Imm) + (PC + 8)$$

$$Ex + Imm = \text{Sign Extended } (imm\ 24) \times 4$$



Data Processing Instruction



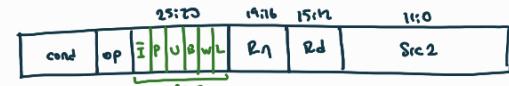
Imm Src

00 : Zero-extended imm8 (data-process)

01 : Zero-extended imm12 (LDRI STR)

10 : Sign-Extended imm24 (multiplied by 4 for B)

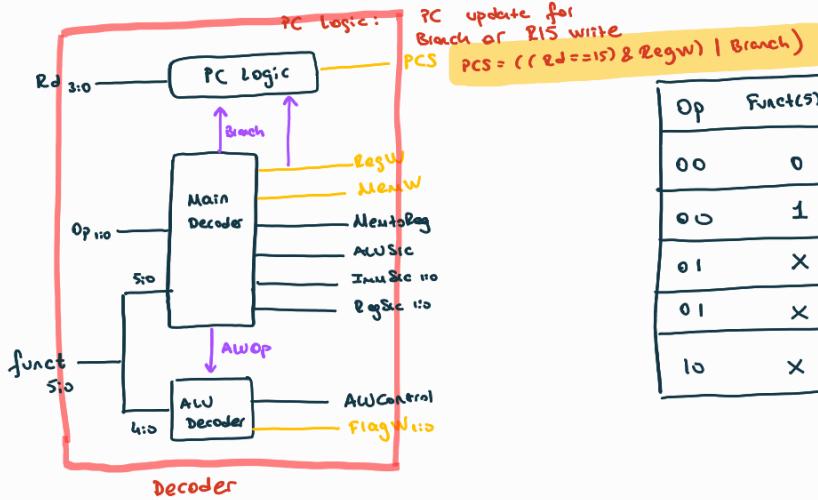
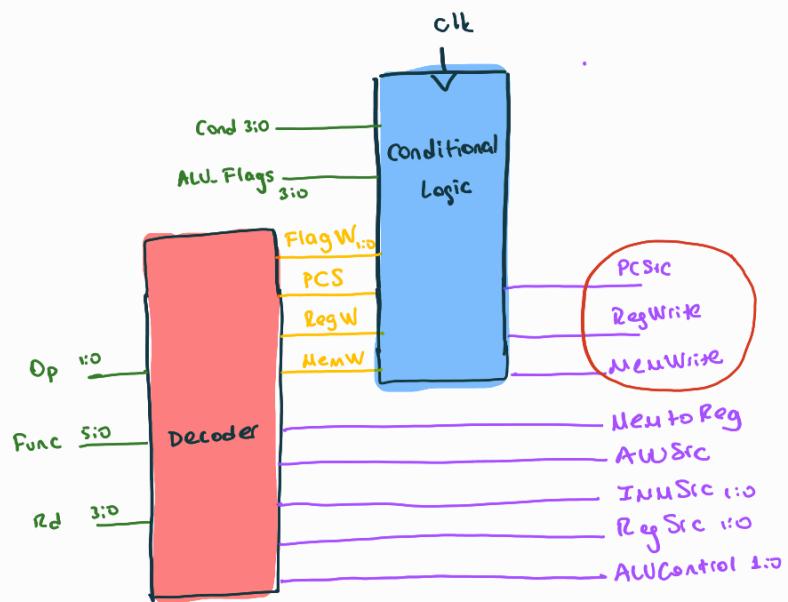
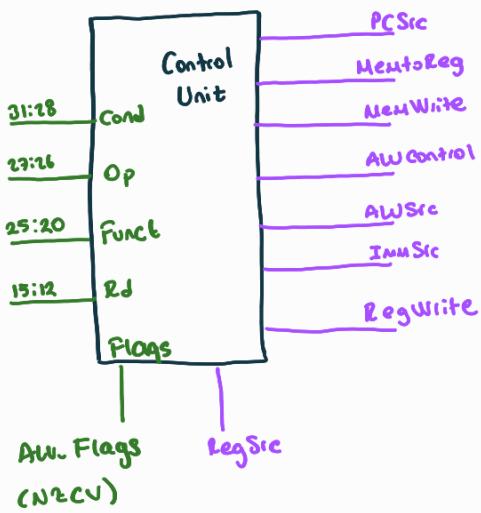
Memory Instruction



Branch Instruction



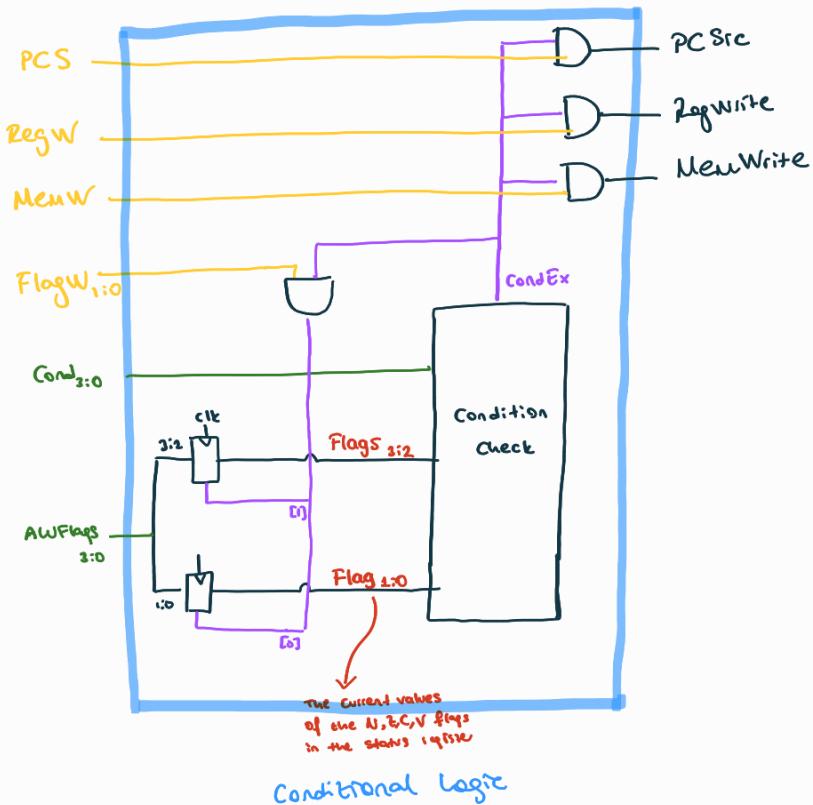
Control Unit



Main Decoder Signal Outputs										
Op	Func(5)	Func(0)	Type	Branch	MemtoReg	NewW	INNSrc	RegW	RegSrc 1:0	AWControl 1:0
00	0	X	Data Reg	0	0	0	0	xx	1 00	1
00	1	X	Data Imm	0	0	0	1	00	1 0x (imm)	1
01	X	0	STR	0	X	1	1	01	0 1	0
01	X	1	LDR	0	1	0	1	01	1 0x (imm)	0
10	X	X	B	1	0	0	1	10	0 1 X 0	0

ADD, SUB update all flags (N2CV)
AND, ORR only update N2 flags

FlagW₁ = 1 → N2 saved
FlagW₀ = 1 → CU saved



<i>ALUOp</i>	<i>Funct_{4:1} (cmd)</i>	<i>Funct₀ (S)</i>	Type	<i>ALUControl_{1:0}</i>	<i>FlagW_{1:0}</i>
0	X	X	Not DP	00 (Add)	00
1	0100	0	ADD	00 (Add)	00
		1			11
	0010	0	SUB	01 (Sub)	00
		1			11
	0000	0	AND	10 (And)	00
		1			10
	1100	0	ORR	11 (Or)	00
		1			10

Digital Design and Computer Architecture, Harris

cond	Mnemonic	Name	CondEx
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 or 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	$\bar{N} \oplus V$
1011	LT	Signed less than	$N \oplus V$
1100	GT	Signed greater than	$\bar{Z}(\bar{N} \oplus V)$
1101	LE	Signed less than or equal	Z OR ($N \oplus V$)
1110	AL (or none)	Always / unconditional	Ignored

ADD, SUB update all flags (N Z CV)
 AND, ORR only update NZ flags

Flag W₁ = 1 → NZ saved
 Flag W₀ = 1 → CV saved

Question 1

$$PC = 0xA184$$

$$CPSR_{31:28} = \boxed{1001}$$

and the ARM assembly code:

0xA184 BNI THERE
0xA194 THERE

Branch if Nans Condition \rightarrow If $N=1$

$\rightarrow CPSR(NZCV) = 1001$

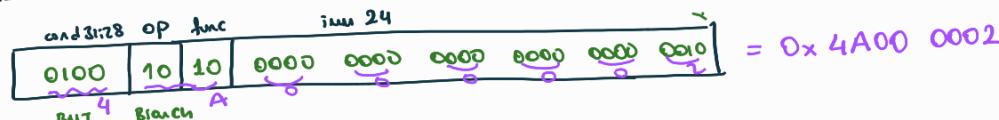
Thus, condition will be executed

$CondEx = 1$

- a) Construct the Machine Code for the instruction
 b) Show the values of all signals in the control and datapath
 c) What are the values of the following after the instruction is executed in the next clock cycle. (PC, CPSR, R15=?)

a) Machine Code:

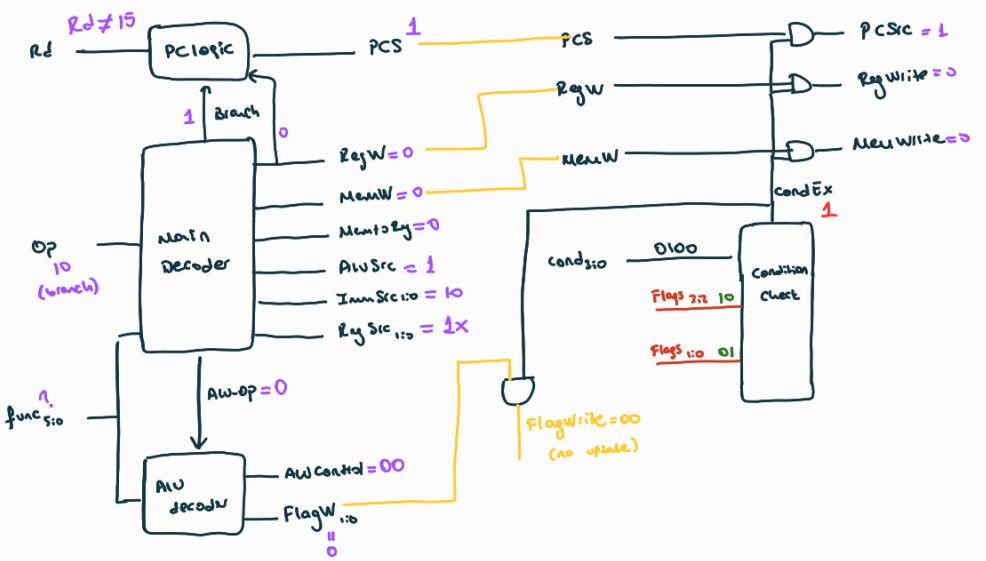
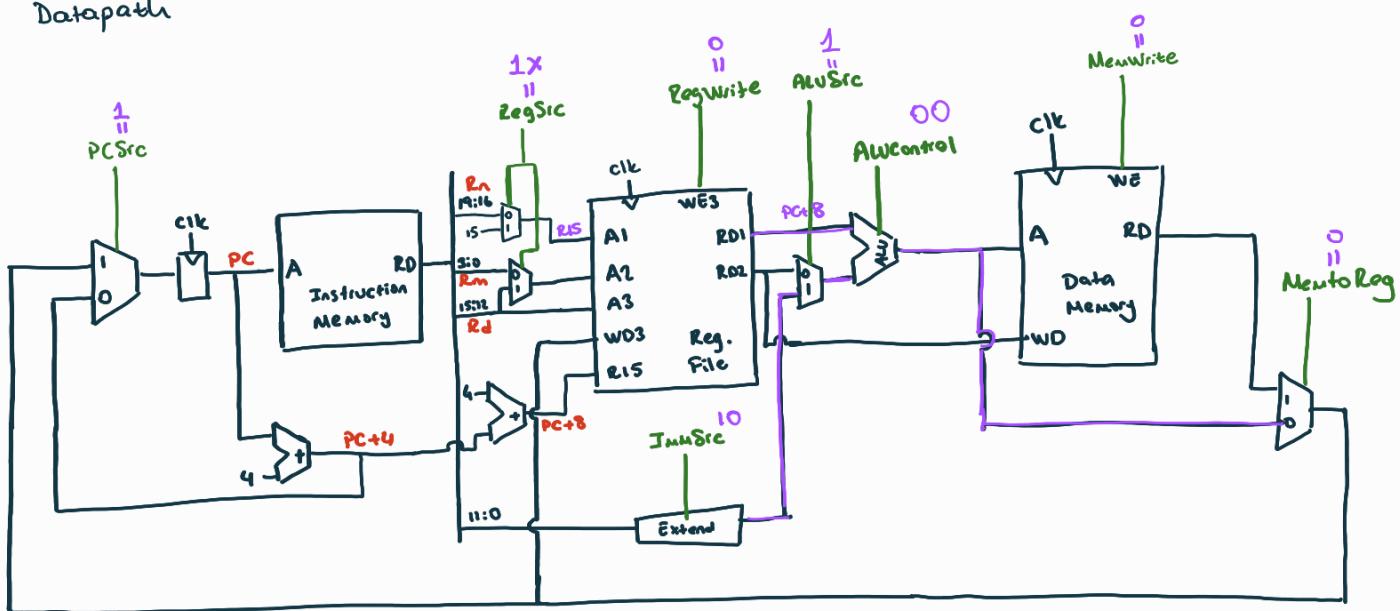
Branch



$$BTA = PC = 0xA194 = (\text{ExtImm}) + (PC+8)$$

$$\text{ExtImm} = \underbrace{\text{Sign Extended imm24}}_2 \ll 2 (x4) = 0xA194 - 0xA18C = 16(\underbrace{9-8}_1) + \underbrace{4-C}_8 = 8$$

b) Datapath



- c) In the next clock cycle
 $PC = BTA = 0xA194$
 $CPSR_{31:28} = \boxed{1001}$ (no change)
 $R15 = PC+8 = 0xA19C$

Question 2

Given:

$$PC = 0xA100$$

$$CPSR_{21:28} = 1100$$

$$R9 = 0x0000$$

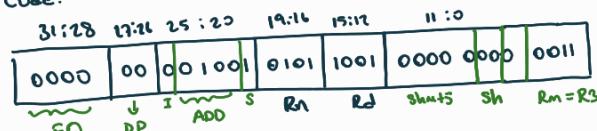
$$R5 = 0x1EDC$$

$$R3 = 0x0000$$

And the ARM machine code in HEX:

$$0xA100 \quad 0x00\ 959\ 003$$

a) Machine Code:

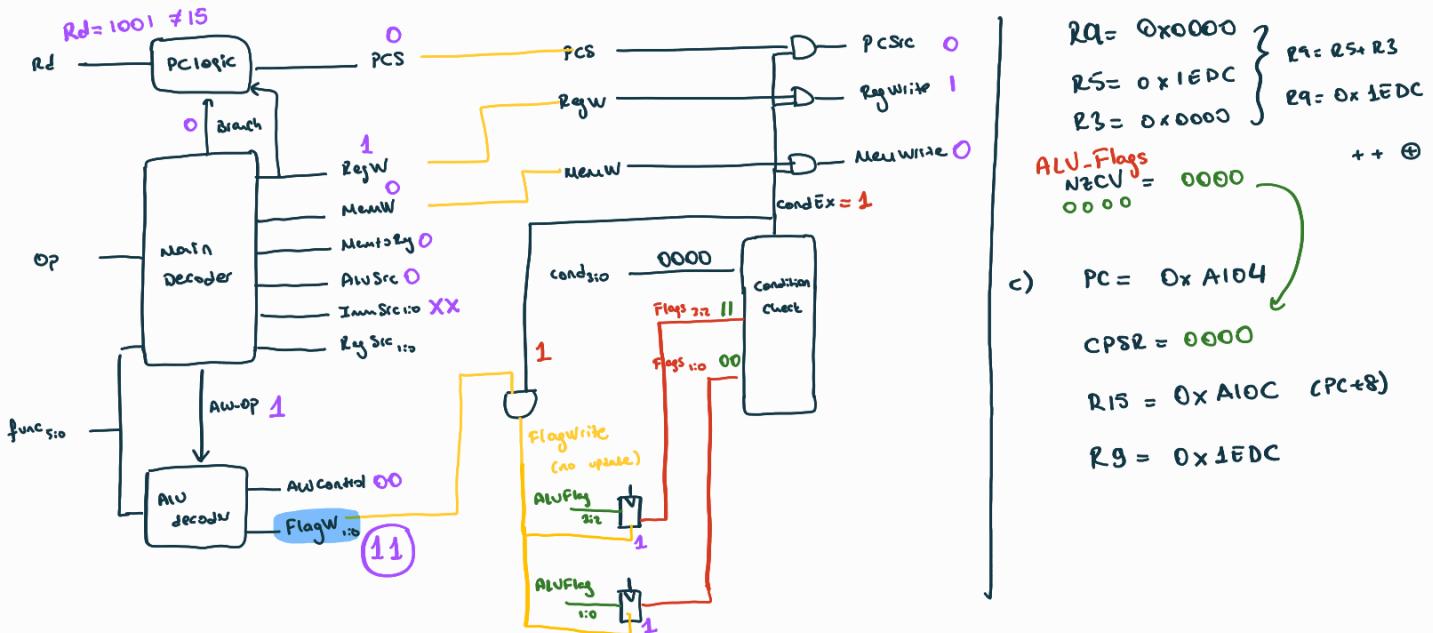
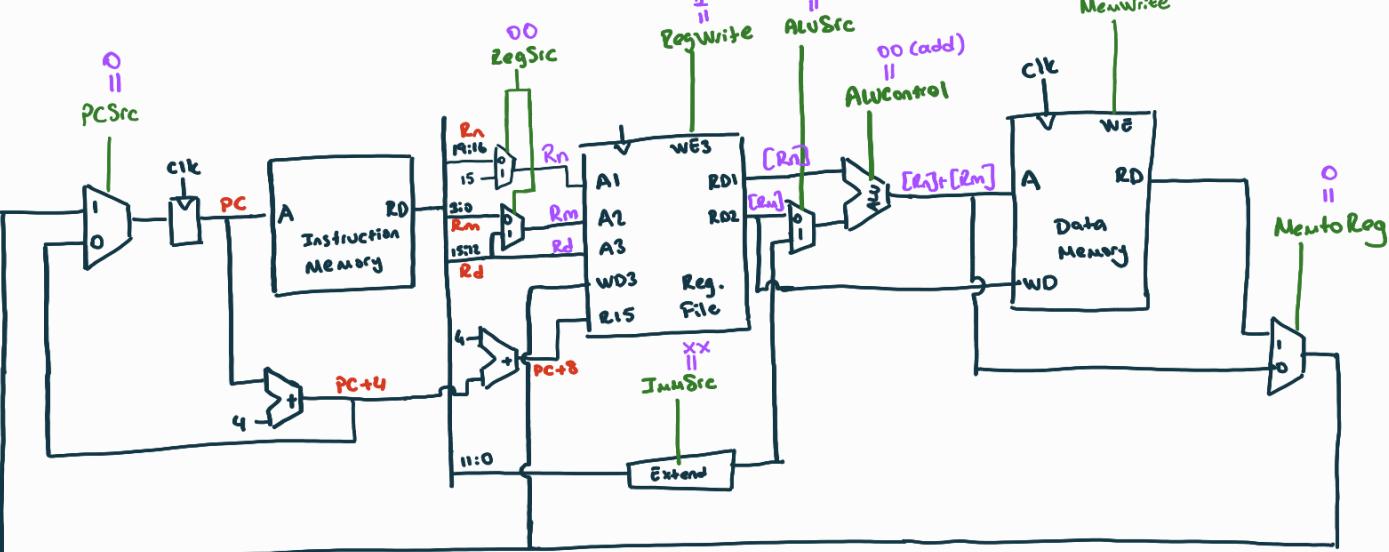


$ADD EQS R9, R5, R3$

$$[R9] \leftarrow [R5] + [R3]$$

\hookrightarrow If $t=1 \Rightarrow \text{CondEx}=1 \Rightarrow \text{Instruction will be executed}$

$$CPSR = 1100 \\ (\text{NzCV})$$



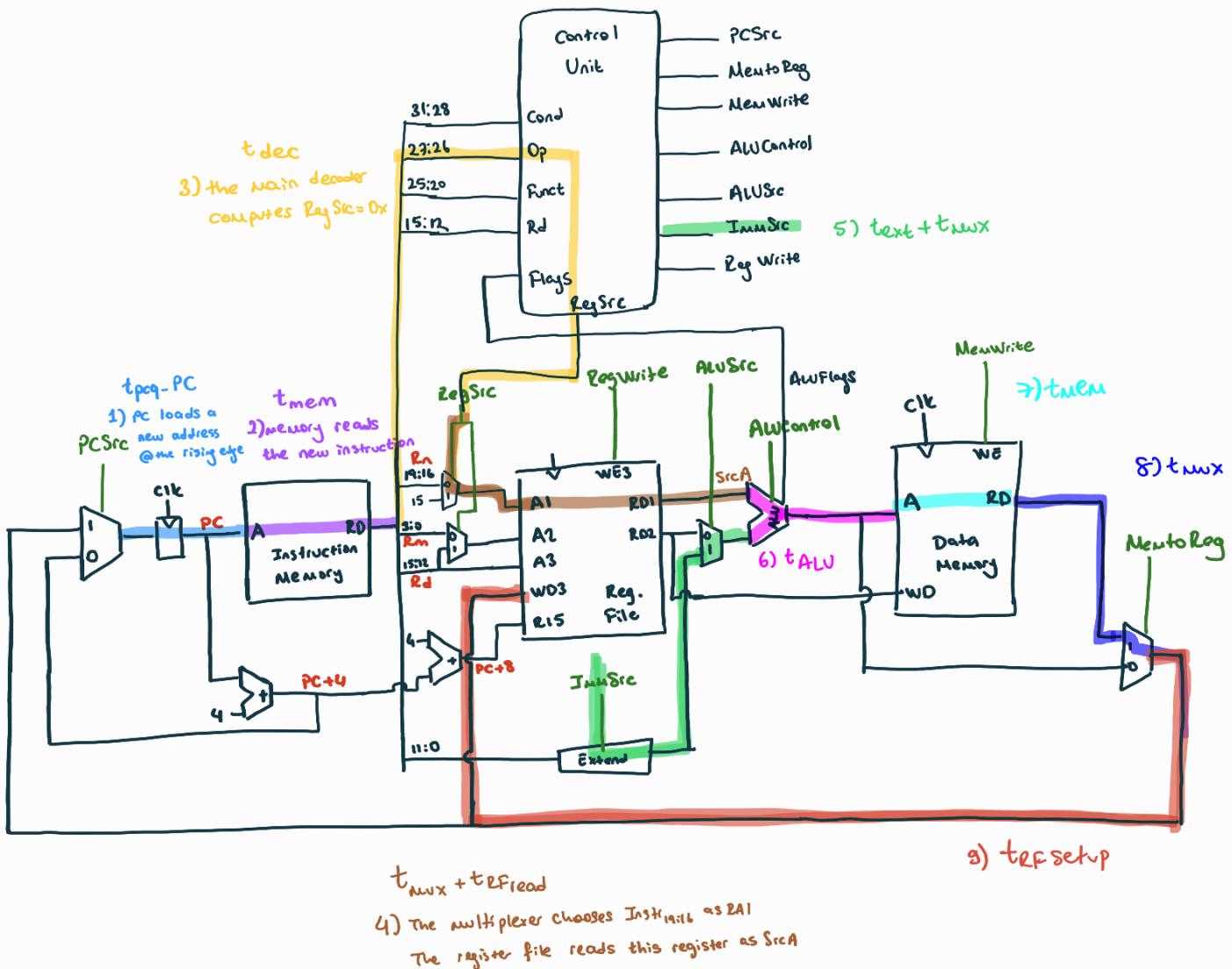
Single Cycle Performance

CPI: Cycles / instruction = 1 (for single-cycle)

clock period (T_c): seconds / cycle

$$\text{Execution Time} = \# \text{ instructions} \times \frac{\text{CPI}}{1} \times T_c$$

Critical Path $\rightarrow T_c$ limited by critical path (LDR)



$$T_{CI} = t_{pcq-PC} + t_{mem} + t_{dec} + \max [t_{NUX} + t_{RF read}, \text{text} + t_{NUX}] + t_{ALU} + t_{Mem} + t_{NUX} + t_{RF setup}$$

\downarrow
 $t_{RF read} > \text{text}$

$$T_{CI} = t_{pcq-PC} + 2t_{mem} + t_{dec} + t_{RF read} + t_{ALU} + 2t_{NUX} + t_{RF setup}$$

* The clock period is constant and must be long enough to accommodate the slowest instruction.

Question 3

- Assume that we have an enhancement to a computer that improves some mode of execution by a factor of 10.

- Enhanced mode is used 50% of the time, measured as a percentage of the execution time when the enhanced mode is in use.

What is the speedup we have obtained from fast mode?

- a) What is the speedup we have obtained from fast mode?
- b) what percentage of the original execution time has been converted to fast mode?

a) New execution time: 100 seconds = 50 sec (old) + 50 sec (new)

Old execution time: 50 sec (old) + $50 \times 10 = 500$ sec = 550 sec

Speed-up = $550 \text{ sec} / 100 \text{ sec} = \boxed{5.5}$

b) $1 - x = \frac{x}{10} \Rightarrow 10 - 10x = x \quad 11x = 10 \rightarrow \boxed{x = 0.91}$