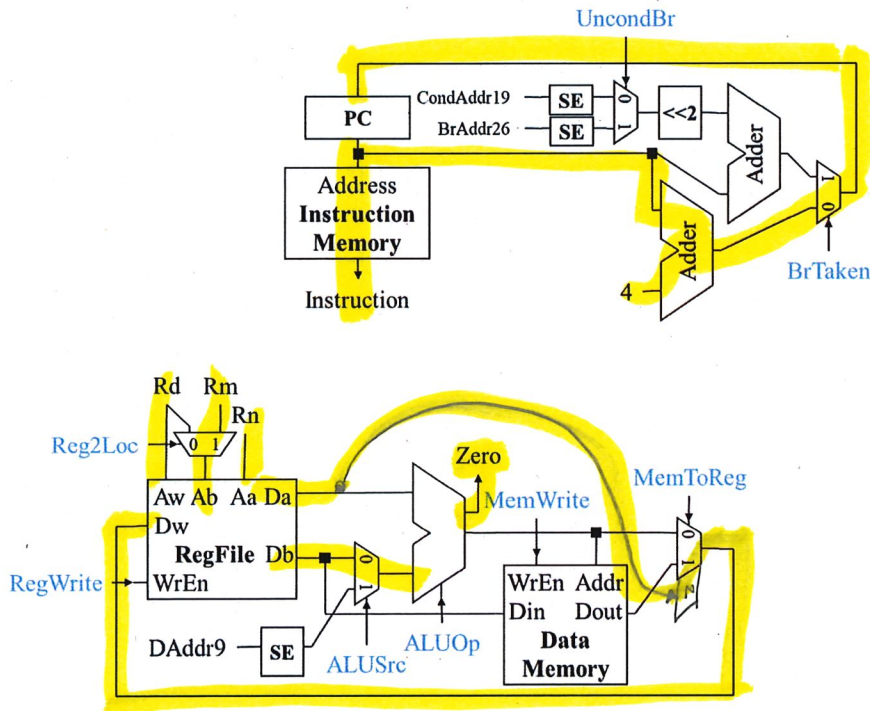


CMOVZ Rd, Rn, Rm: conditional move if zero. The instruction copies the value from the Rn register to the Rd register if the contents of the Rm register are 0. If the contents of the Rm register are non-zero, the rd register is left as-is.

```

instruction = Mem[PC];
cond = (Reg[Rm] == 0);
if (cond)
    Reg[Rd] = Reg[Rn];
PC = PC + 4;
    
```

Reg2Loc	1
ALUSrc	0
MemToReg	2
RegWrite	(Zero)
MemWrite	0
BrTaken	0
UncondBr	X
ALUOp	B == 0?

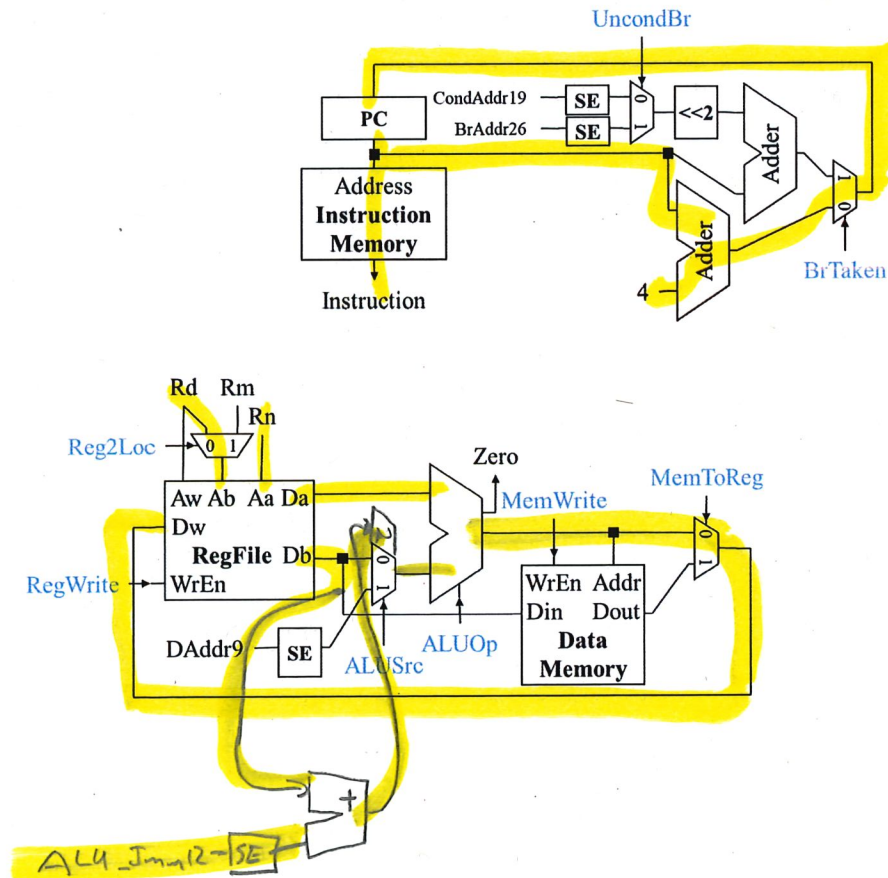


ADD3 Rd, Rn, ALU_Imm12: add together the values of register Rd, register Rn, and the imm12 (interpreted as a 2's comp number), and store the result in register Rd.

instruction = Mem[PC];
Reg[Rd] = Reg[Rn] + Reg[Rn] +
SE(ALU_Imm12);

PC = PC + 4;

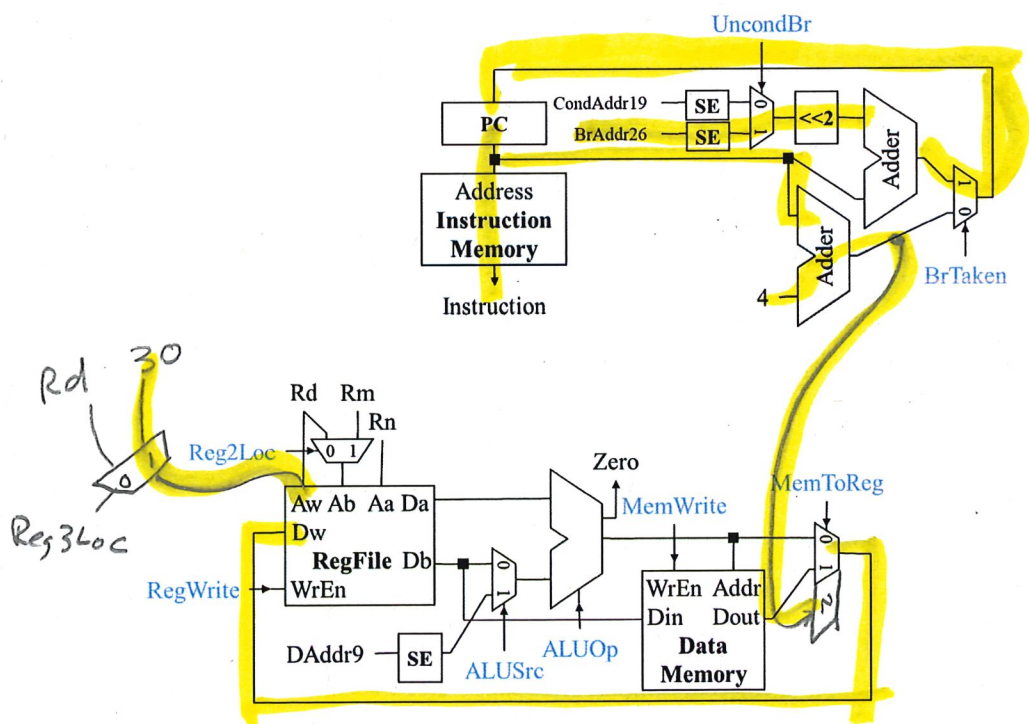
Reg2Loc	0
ALUSrc	2
MemToReg	0
RegWrite	1
MemWrite	0
BrTaken	0
UncondBr	X
ALUOp	+



BL BrAddr26: Same as B, but also stores the value of PC+4 (where PC is the address of this instruction) into register X30.

Instruction = Mem[PC];
 Reg[X30] = PC + 4;
 PC = PC + SE(BrAddr26) << 2;

Reg2Loc	X
ALUSrc	X
MemToReg	2
RegWrite	1
MemWrite	0
BrTaken	1
UncondBr	1
ALUOp	X



SWAP Rd, Rn: An R-type instruction. Takes the values from Rd and Rn and puts them into registers Rn and Rd respectively, thus swapping their contents.

Instruction = Mem[PC];

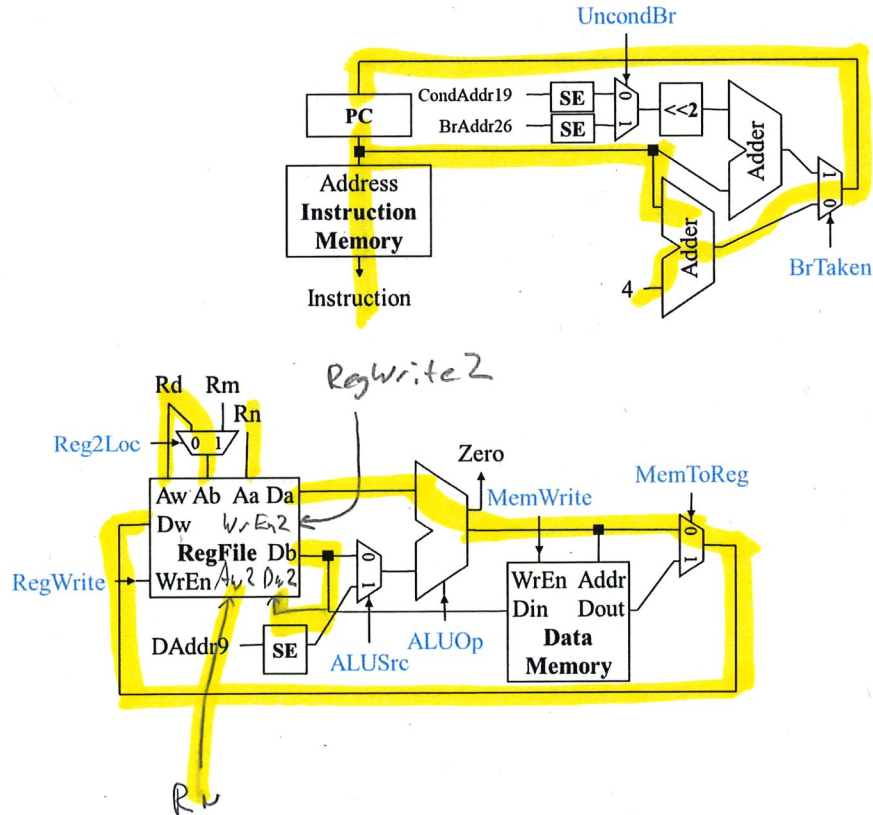
Simultaneously:

Reg[Rd] = Reg[Rn] +

Reg[Rn] = Reg[Rd]

PC = PC + 4;

Reg2Loc	0
ALUSrc	X
MemToReg	0
RegWrite	1
MemWrite	0
BrTaken	0
UncondBr	X
ALUOp	Bypass A
RegWrite2	1



Create a single-cycle CPU that can perform the following two instructions ONLY. Your CPU should be as simple, and as fast, as possible. If anything is not required for this CPU, do not include it. Show both the datapath and the control.

CBNZ Rd, CondAddr19 if (Reg[Rd] != 0) PC = PC + SE(CondAddr19) << 2;

STUR Rd, [Rn, DAddr9] Mem[Reg[Rn] + SE(DAddr9)] = Reg[Rd];

CBNZ: Instruction = Mem[PC];
 Cond = (Reg[Rd] == 0);
 if (!Cond)
 PC = PC + SE(CondAddr19) << 2;
 else
 PC = PC + 4

STUR: Instruction = Mem[PC];
 Addr = Reg[Rn] + SE(DAddr9);
 Mem[Addr] = Reg[Rd];
 PC = PC + 4

	CBNZ	STUR
ALUSrc	0	1
MemWrite	0	1
BrTaken (!zero)	0	0
ALUOp B==0?	+	+

