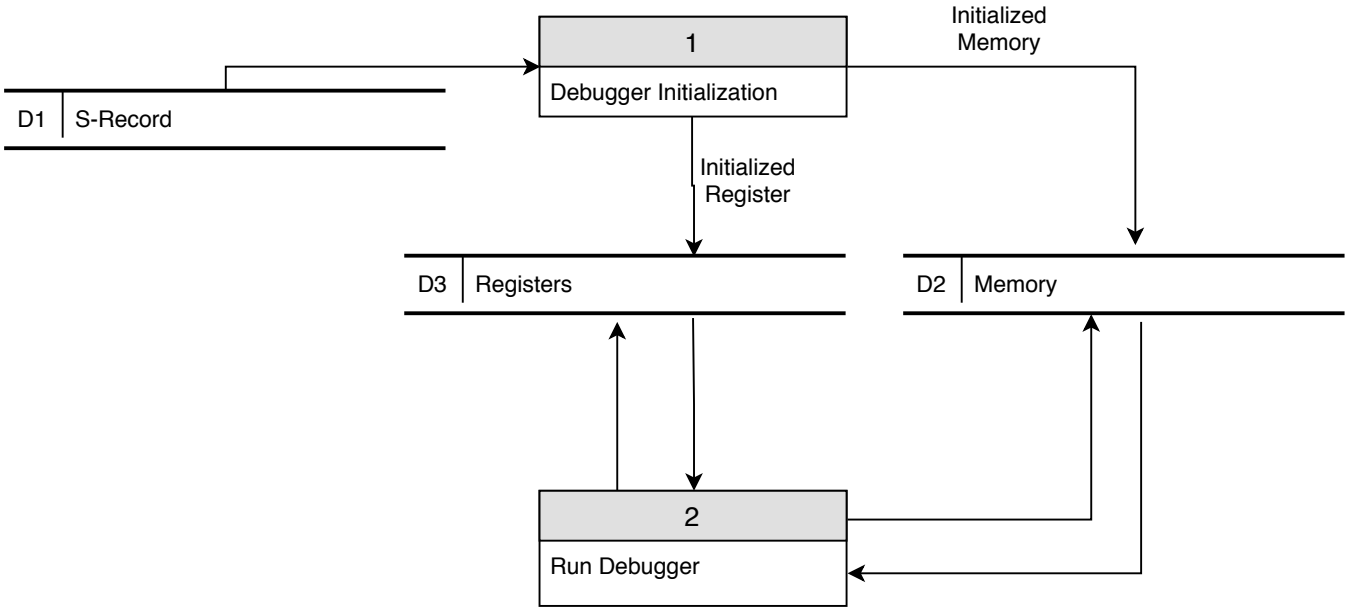
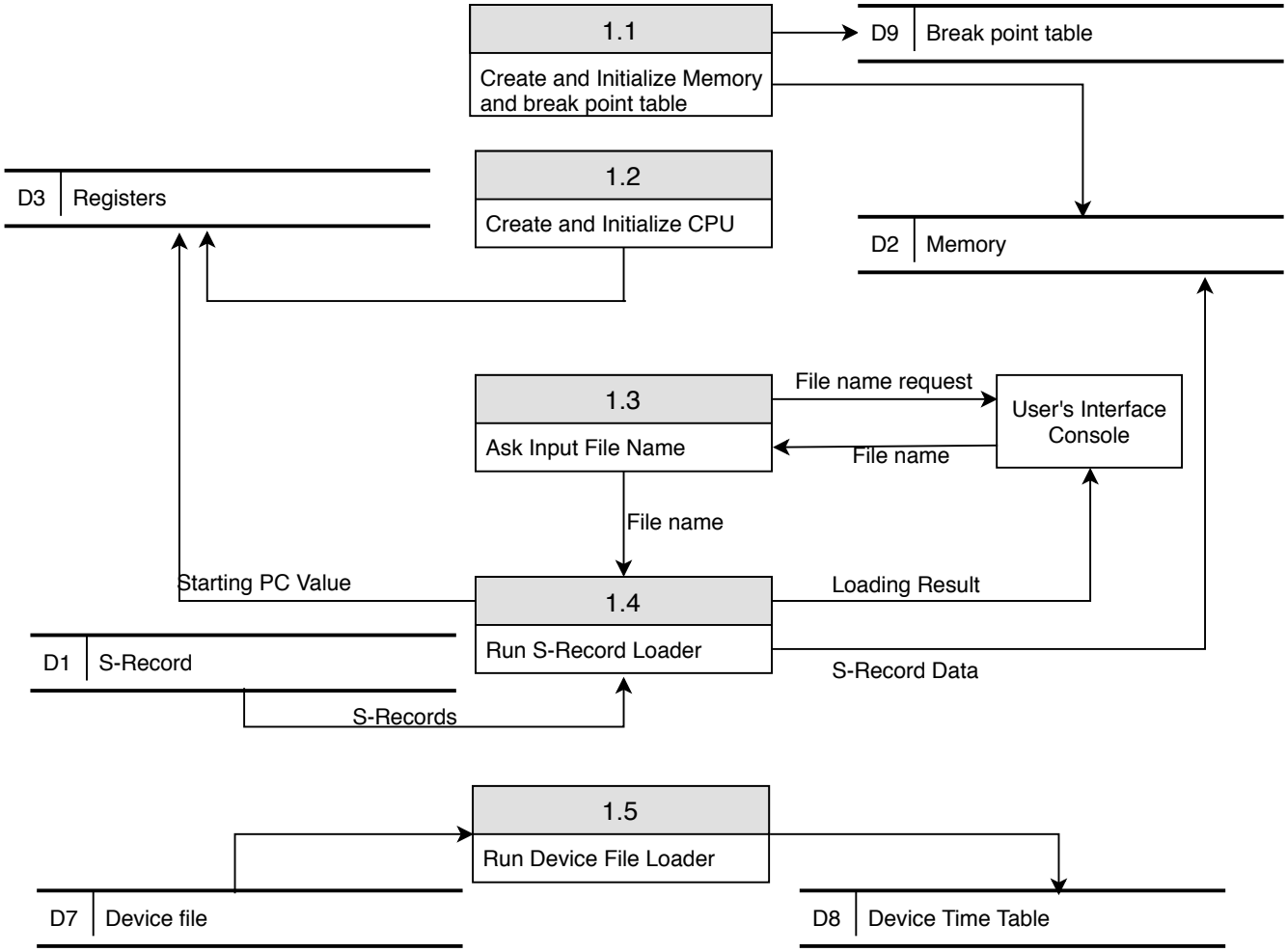
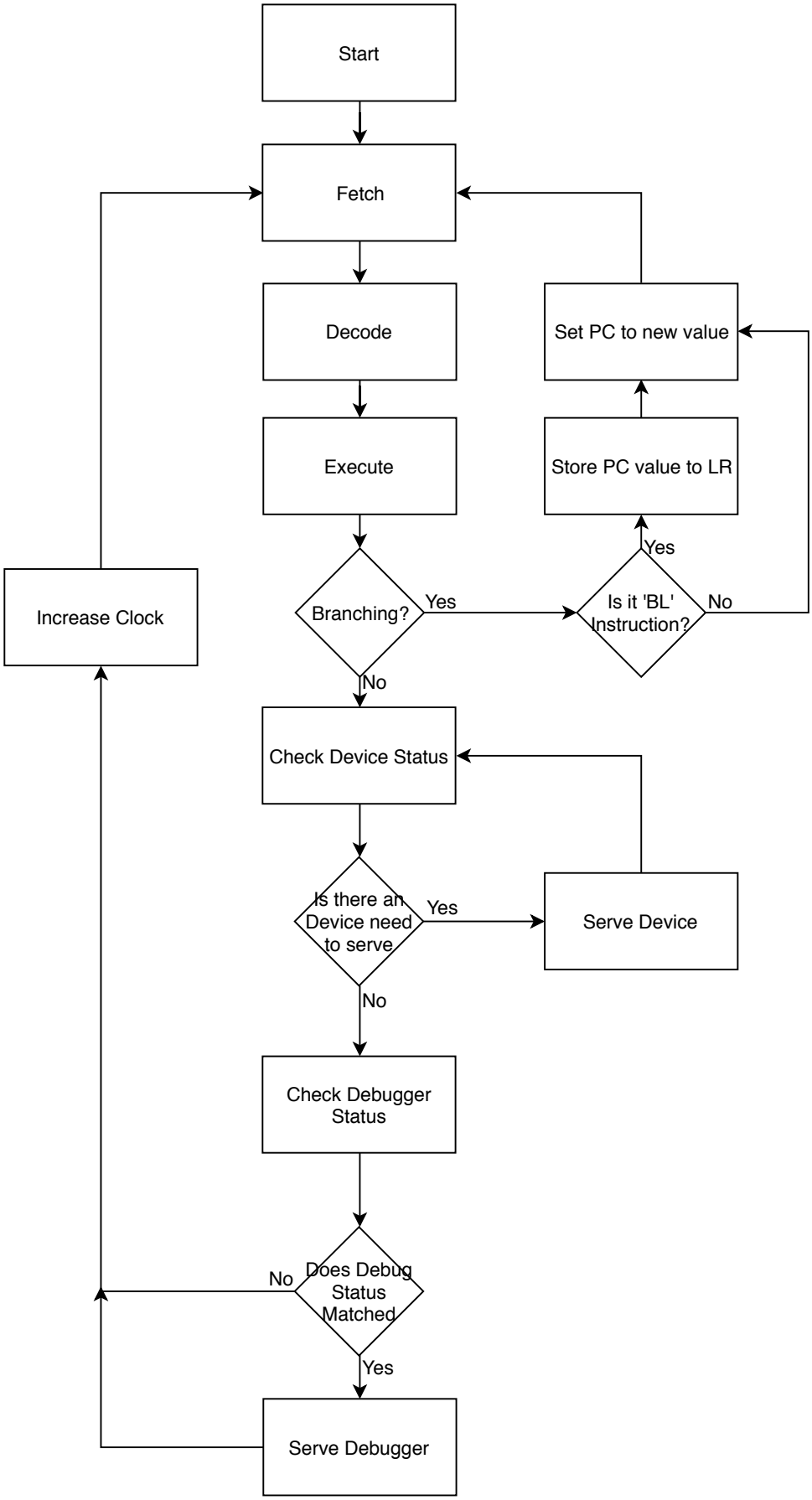
Level 1 Data Flow Diagram



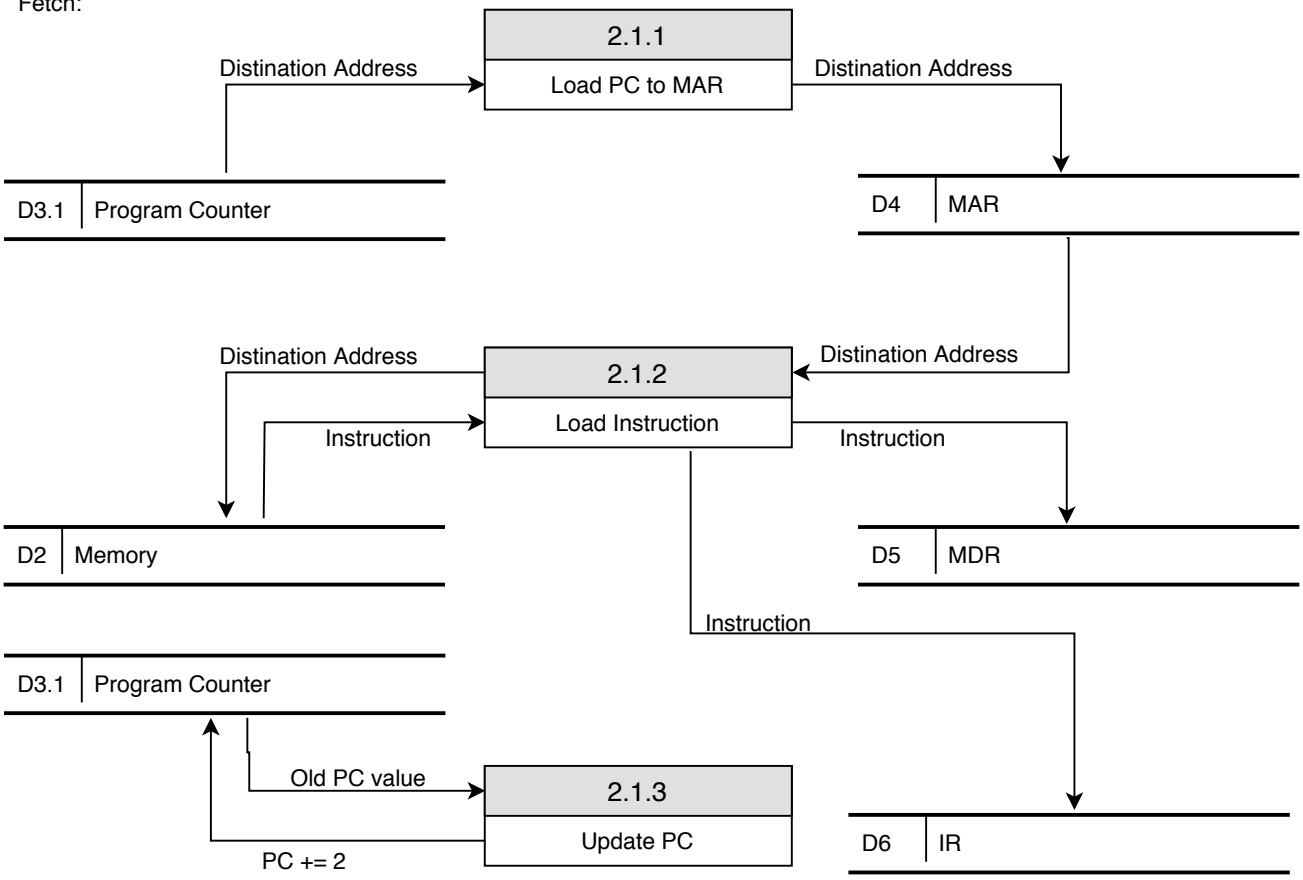
Data Flow Diagram
Debugger Initialization level 2

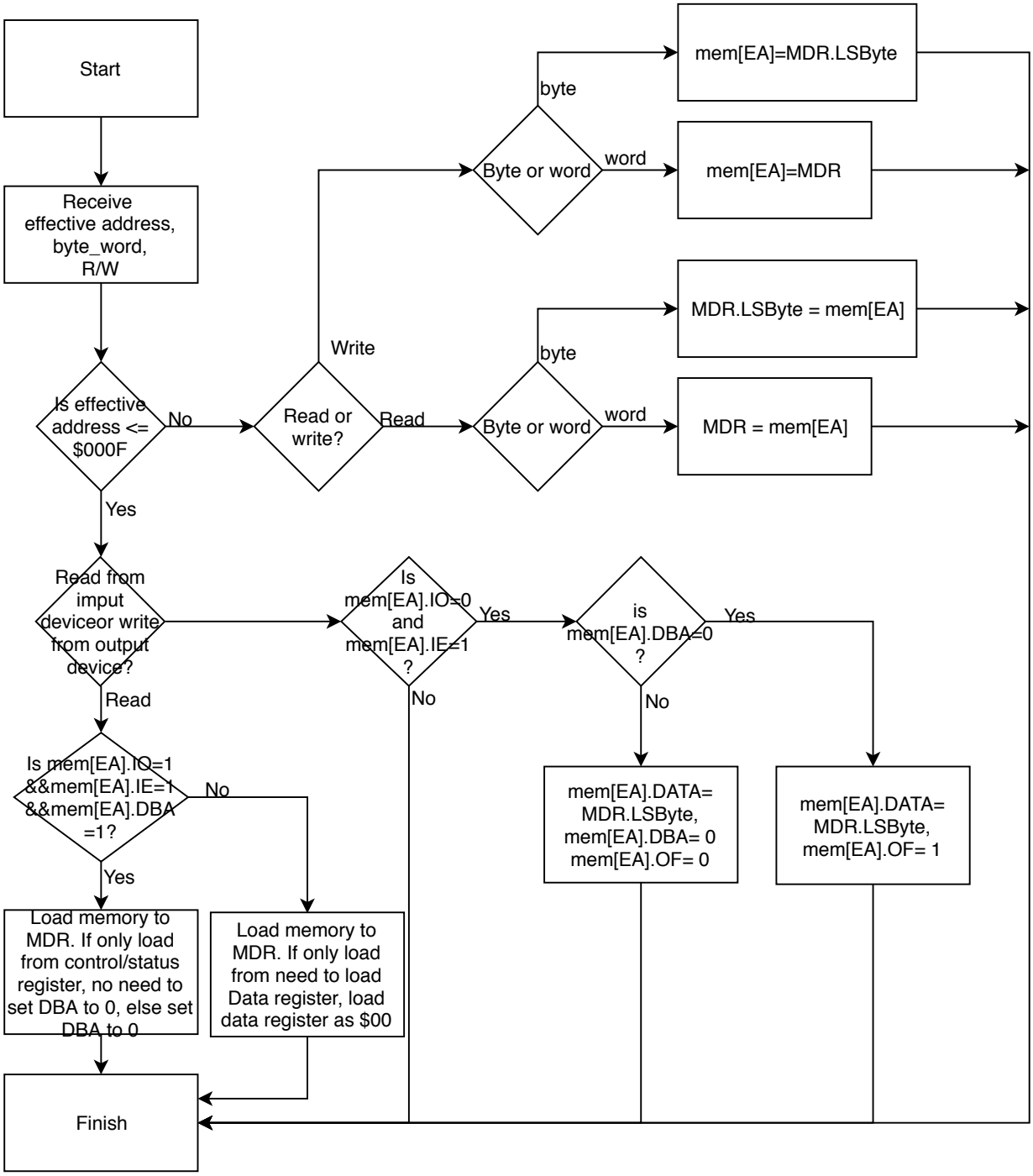


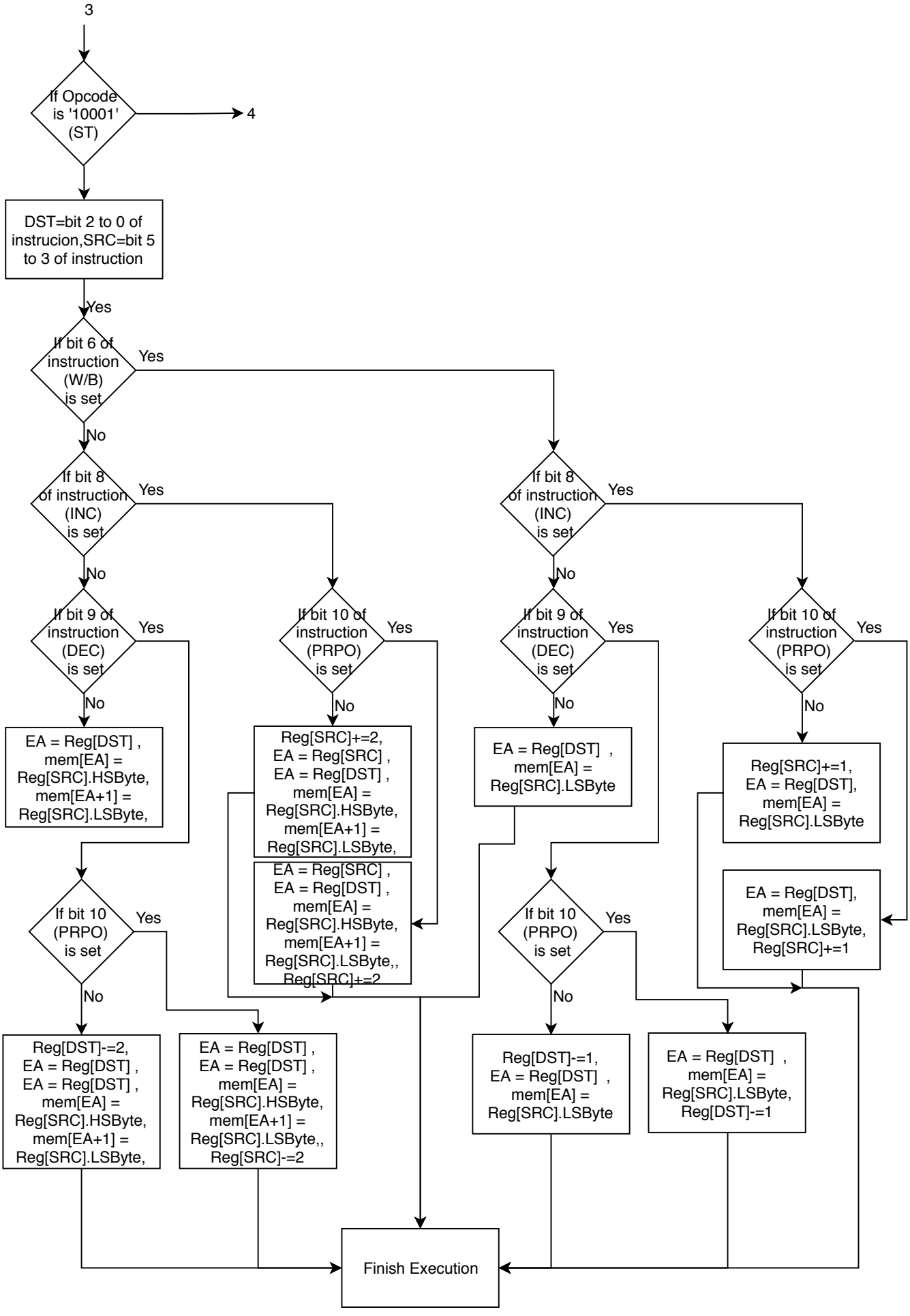
Process Flow Diagram of Run CPU

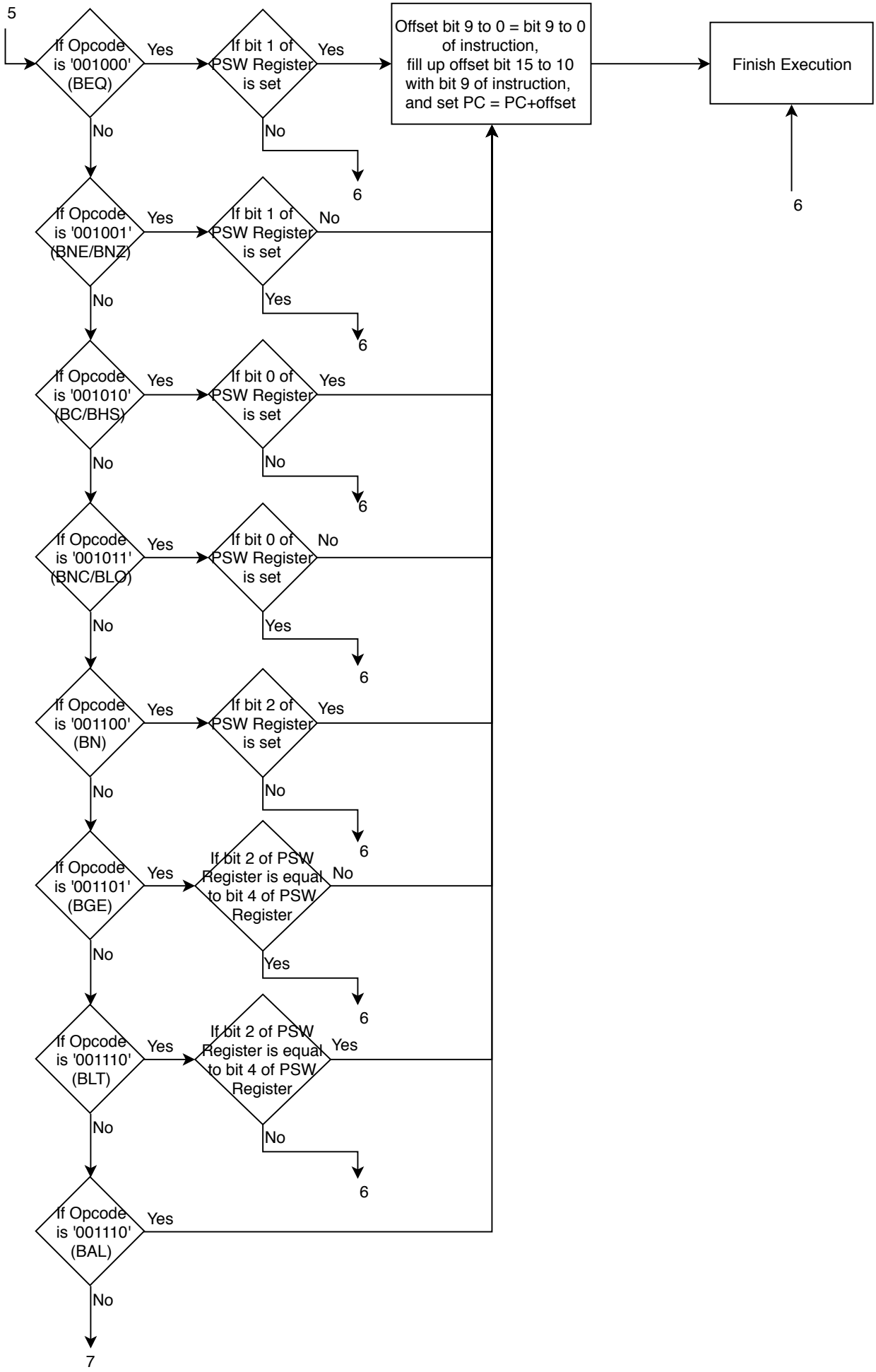


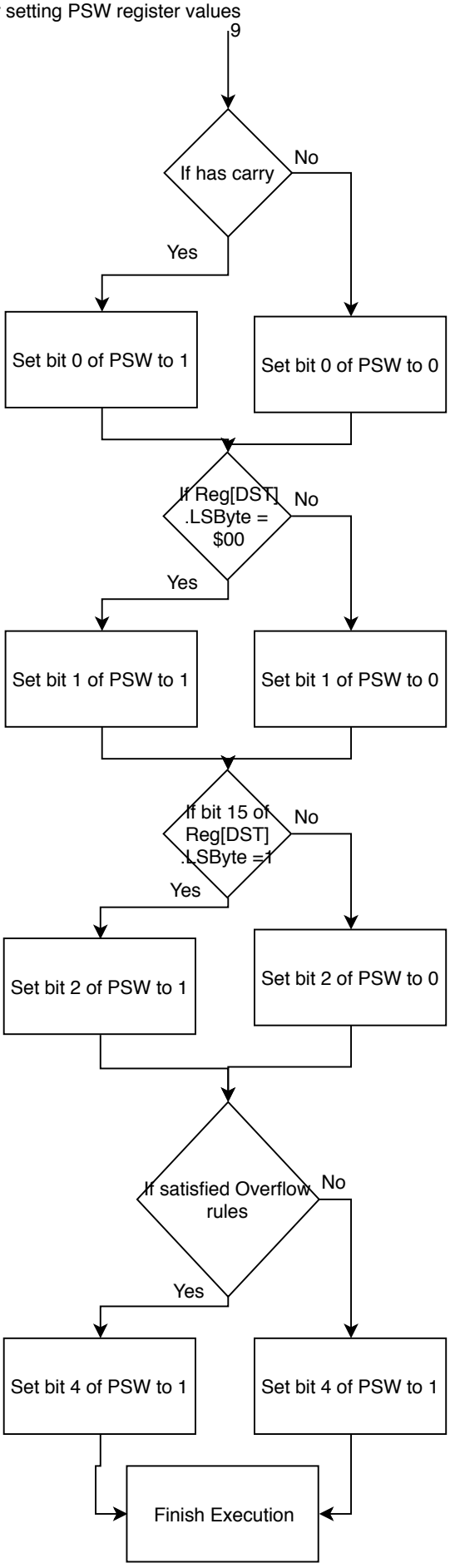
Data Flow Diagram
Fetch:

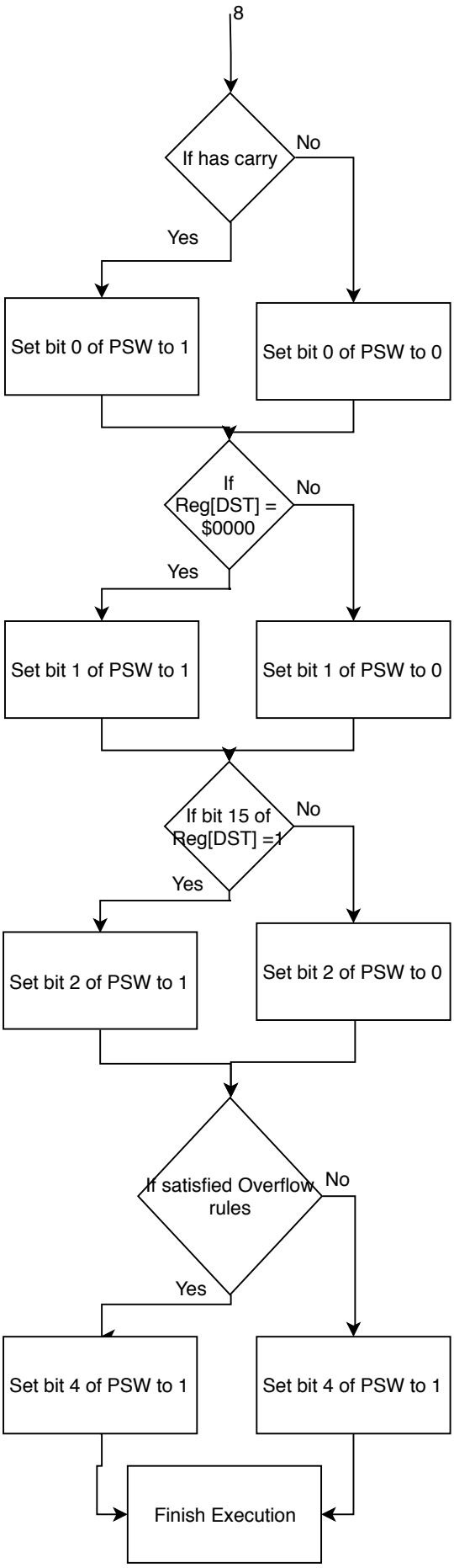



Start
Receive effective address, byte_word, R/W
Is effective address <= $000F
No
Yes
Read or write?
Write
Read
Byte or word
byte
word
mem[EA]=MDR.LSByte
mem[EA]=MDR
Byte or word
byte
word
MDR.LSByte = mem[EA]
MDR = mem[EA]
Read from imput deviceor write from output device?
Read
Is mem[EA].IO=0 and mem[EA].IE=1 ?
Yes
No
is mem[EA].DBA=0 ?
Yes
No
Is mem[EA].IO=1 &&mem[EA].IE=1 &&mem[EA].DBA =1?
No
Yes
mem[EA].DATA= MDR.LSByte, mem[EA].DBA= 0 mem[EA].OF= 0
mem[EA].DATA= MDR.LSByte, mem[EA].OF= 1
Load memory to MDR. If only load from control/status register, no need to set DBA to 0, else set DBA to 0
Load memory to MDR. If only load from need to load Data register, load data register as $00
Finish


3
If Opcode is '10001' (ST)
4
DST=bit 2 to 0 of instrucion,SRC=bit 5 to 3 of instruction
Yes
If bit 6 of instruction (W/B) is set
Yes
No
If bit 8 of instruction (INC) is set
Yes
No
If bit 9 of instruction (DEC) is set
Yes
No
EA = Reg[DST] , mem[EA] = Reg[SRC].HSByte, mem[EA+1] = Reg[SRC].LSByte,
If bit 10 of instruction (PRPO) is set
Yes
No
Reg[SRC]+=2, EA = Reg[SRC] , EA = Reg[DST] , mem[EA] = Reg[SRC].HSByte, mem[EA+1] = Reg[SRC].LSByte,
EA = Reg[SRC] , EA = Reg[DST] , mem[EA] = Reg[SRC].HSByte, mem[EA+1] = Reg[SRC].LSByte,, Reg[SRC]+=2
If bit 10 (PRPO) is set
Yes
No
Reg[DST]-=2, EA = Reg[DST] , EA = Reg[DST] , mem[EA] = Reg[SRC].HSByte, mem[EA+1] = Reg[SRC].LSByte,
EA = Reg[DST] , EA = Reg[DST] , mem[EA] = Reg[SRC].HSByte, mem[EA+1] = Reg[SRC].LSByte,, Reg[SRC]-=2
If bit 8 of instruction (INC) is set
Yes
No
If bit 9 of instruction (DEC) is set
Yes
No
EA = Reg[DST] , mem[EA] = Reg[SRC].LSByte
If bit 10 (PRPO) is set
Yes
No
Reg[DST]-=1, EA = Reg[DST] , mem[EA] = Reg[SRC].LSByte
EA = Reg[DST] , mem[EA] = Reg[SRC].LSByte, Reg[DST]-=1
If bit 10 of instruction (PRPO) is set
Yes
No
Reg[SRC]+=1, EA = Reg[DST], mem[EA] = Reg[SRC].LSByte
EA = Reg[DST], mem[EA] = Reg[SRC].LSByte, Reg[SRC]+=1
Finish Execution


5
If Opcode is '001000' (BEQ)
Yes
If bit 1 of PSW Register is set
Yes
Offset bit 9 to 0 = bit 9 to 0 of instruction, fill up offset bit 15 to 10 with bit 9 of instruction, and set PC = PC+offset
Finish Execution
No
No
6
6
If Opcode is '001001' (BNE/BNZ)
Yes
If bit 1 of PSW Register is set
No
No
Yes
6
If Opcode is '001010' (BC/BHS)
Yes
If bit 0 of PSW Register is set
Yes
No
No
6
If Opcode is '001011' (BNC/BLO)
Yes
If bit 0 of PSW Register is set
No
No
Yes
6
If Opcode is '001100' (BN)
Yes
If bit 2 of PSW Register is set
Yes
No
No
6
If Opcode is '001101' (BGE)
Yes
If bit 2 of PSW Register is equal to bit 4 of PSW Register
No
No
Yes
6
If Opcode is '001110' (BLT)
Yes
If bit 2 of PSW Register is equal to bit 4 of PSW Register
Yes
No
No
6
If Opcode is '001110' (BAL)
Yes
No
7

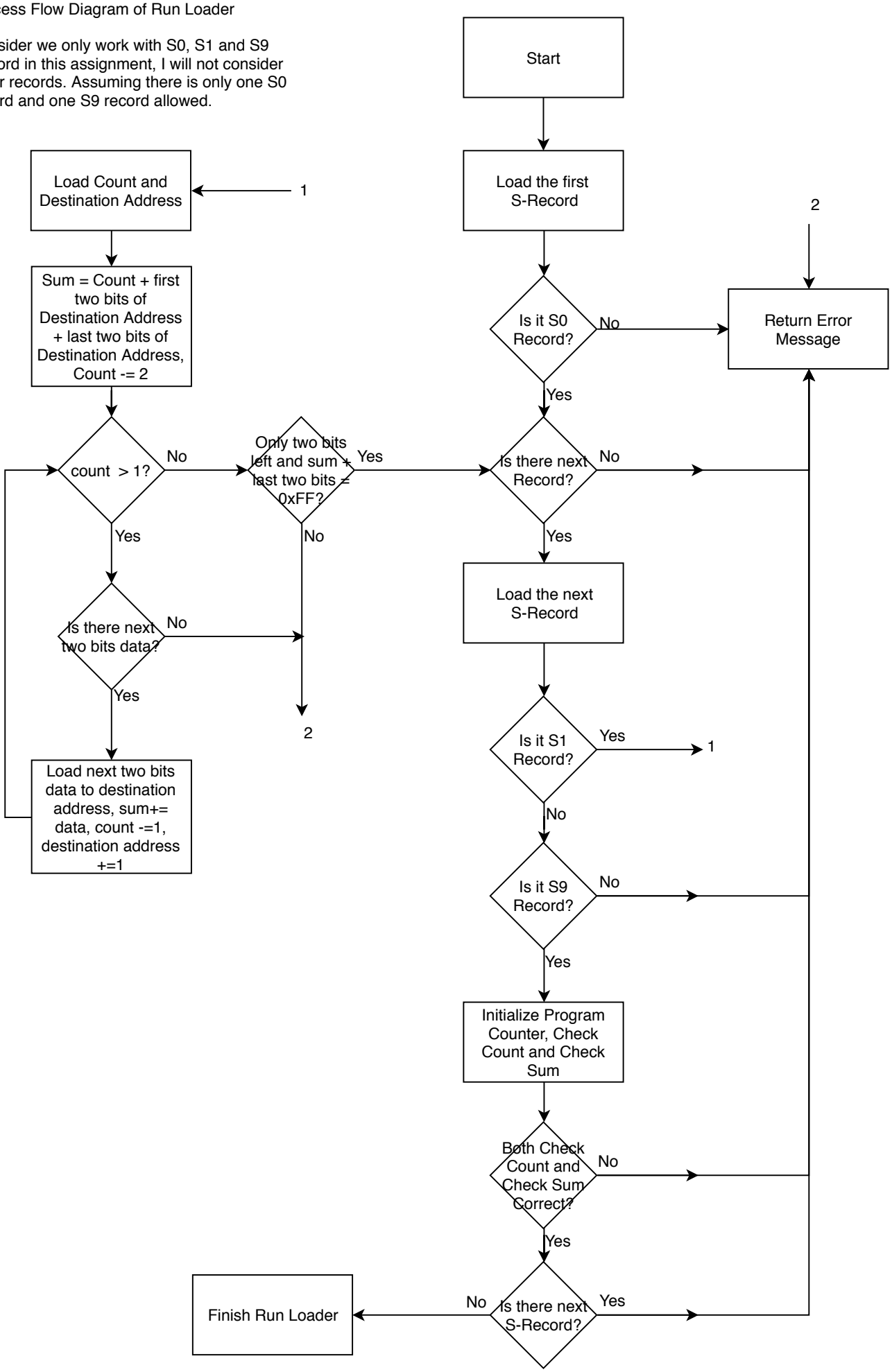


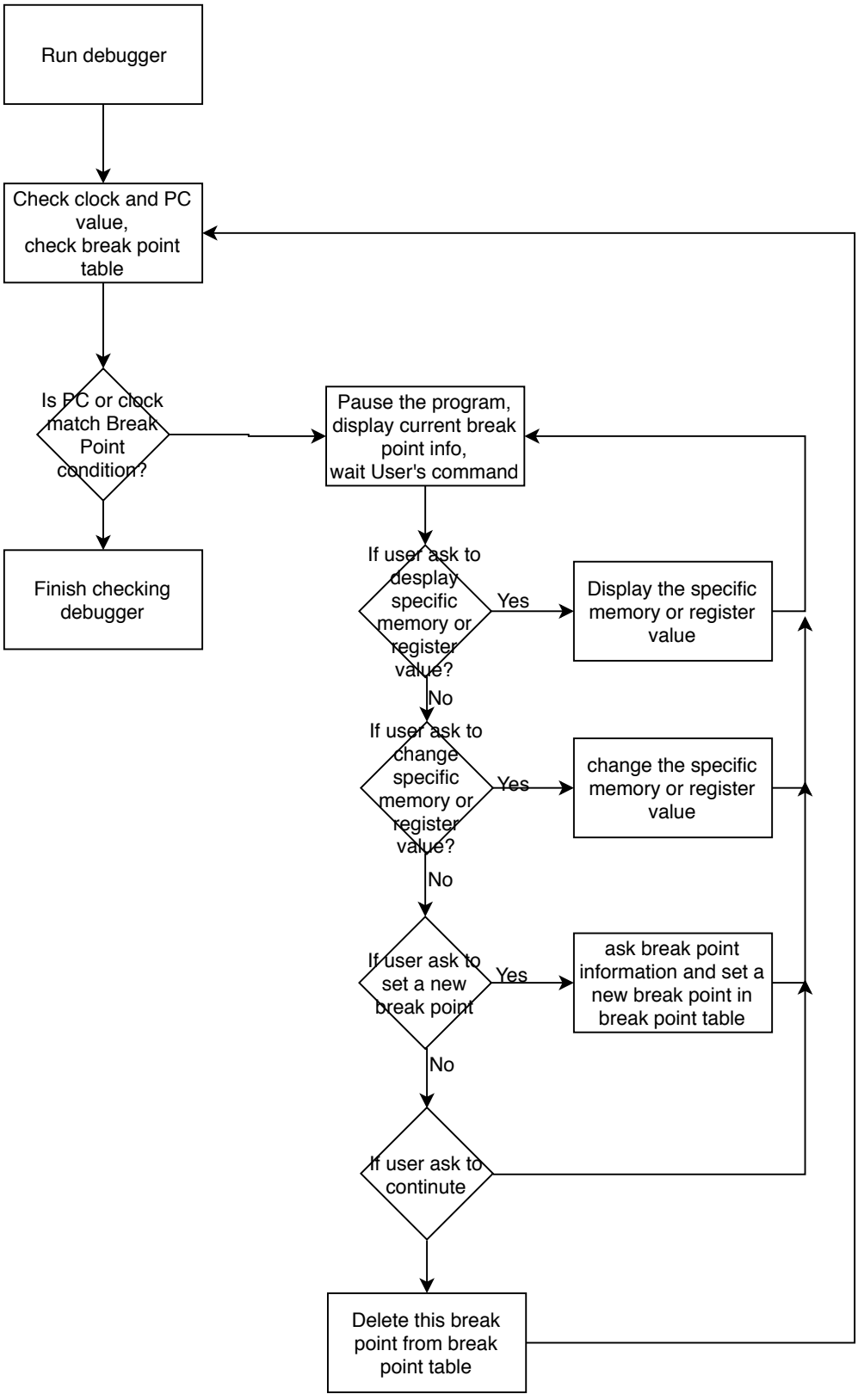


Overflow rules (from data dictionary): -For addition: set overflow if destination sign is same as source sign, but now same as result sign.
-For subtraction: set overflow if destination sign is not same as source sign,
but result sign is same as source sign.

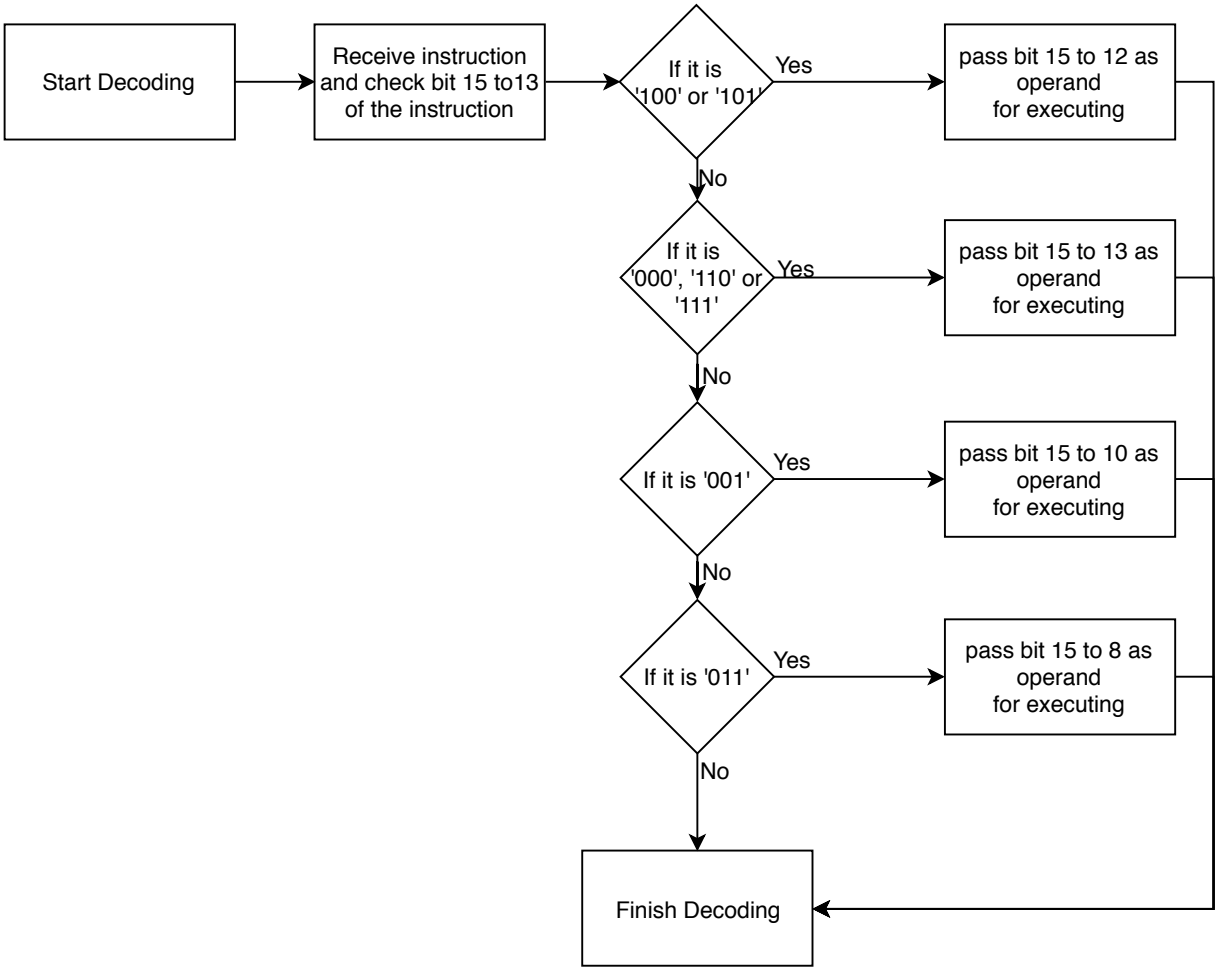
Process Flow Diagram of Run Loader

Consider we only work with S0, S1 and S9 Record in this assignment, I will not consider other records. Assuming there is only one S0 record and one S9 record allowed.

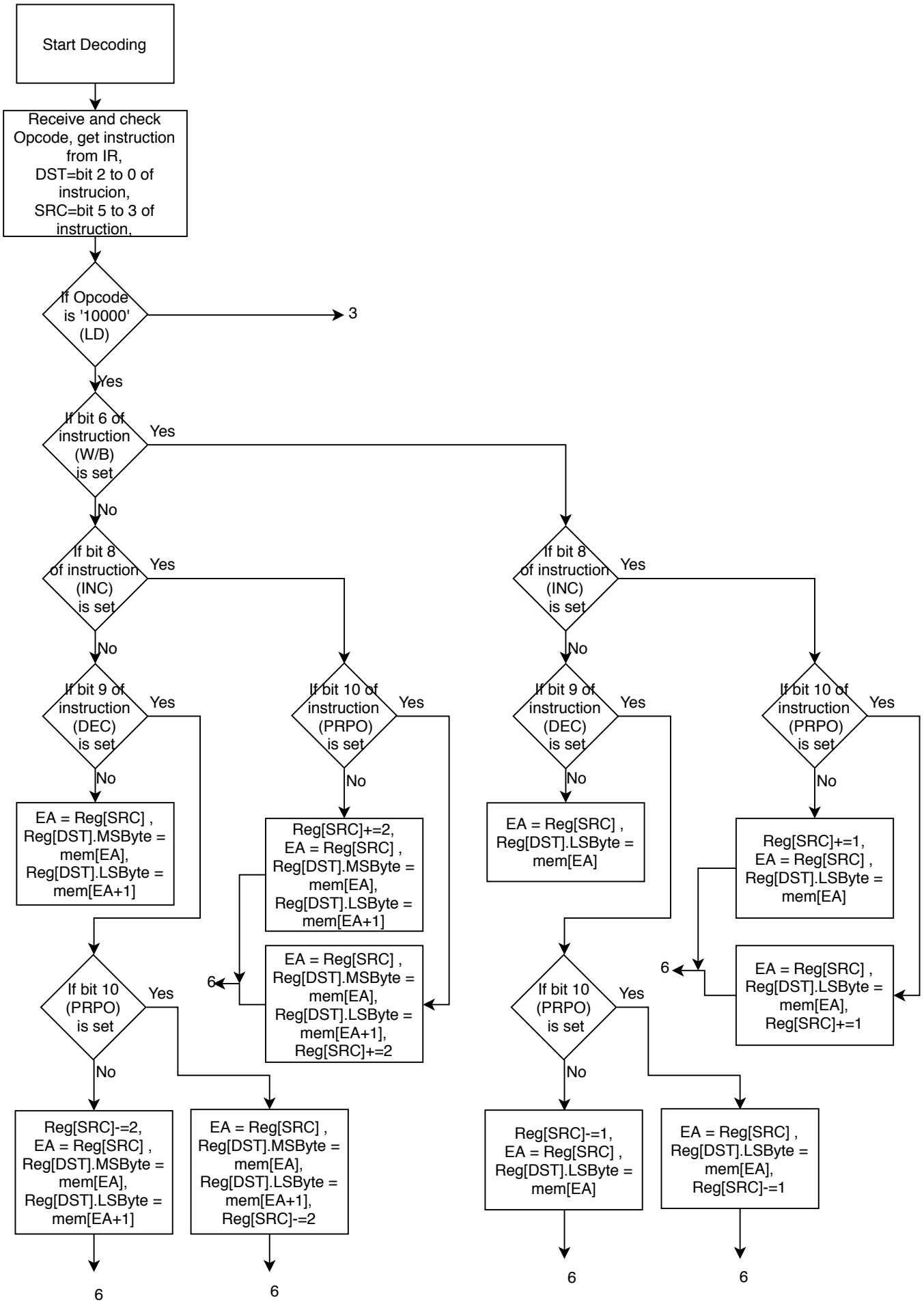


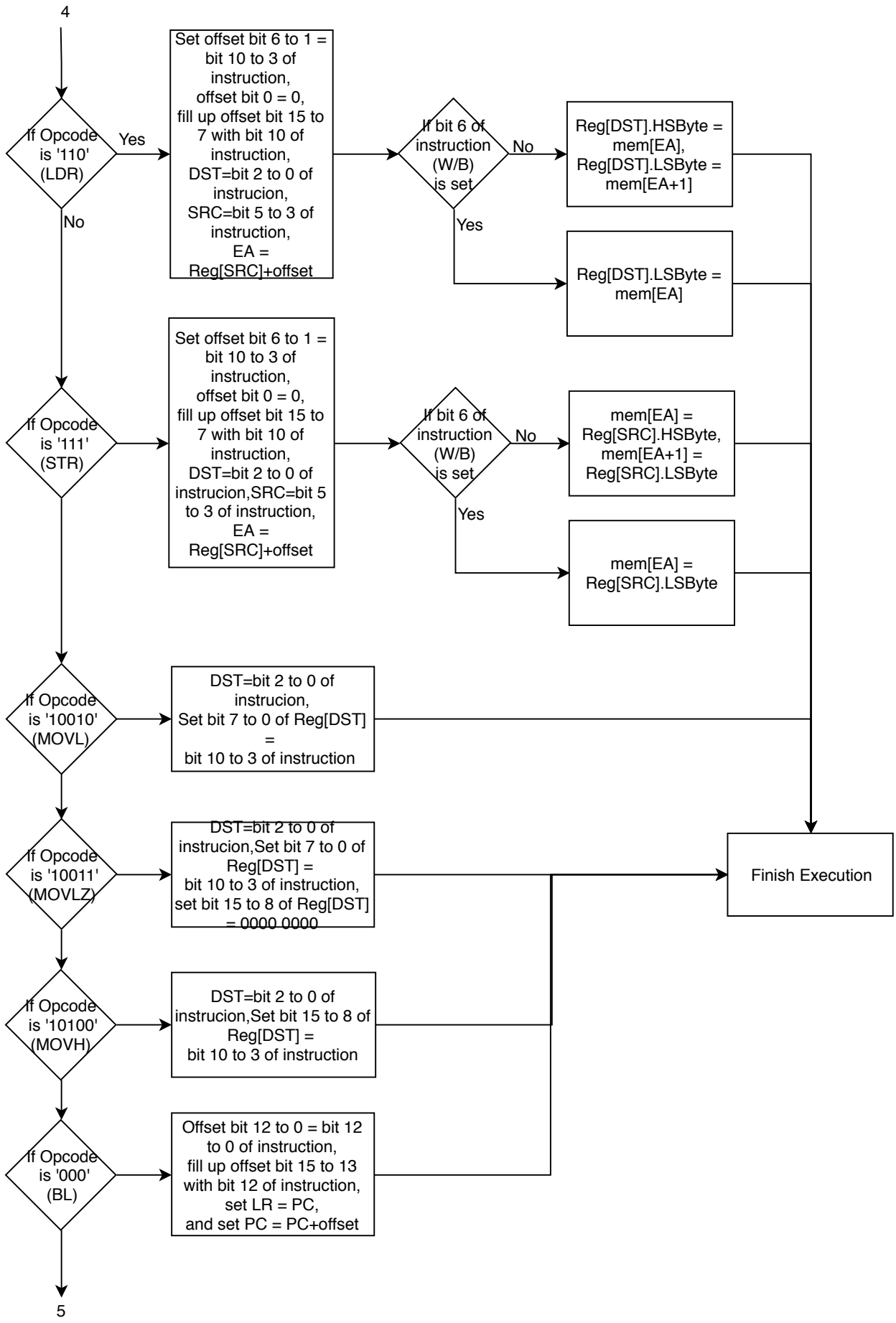

Run debugger
Check clock and PC value, check break point table
Is PC or clock match Break Point condition?
Finish checking debugger
Pause the program, display current break point info, wait User's command
If user ask to desplay specific memory or register value?
Yes
Display the specific memory or register value
No
If user ask to change specific memory or register value?
Yes
change the specific memory or register value
No
If user ask to set a new break point
Yes
ask break point information and set a new break point in break point table
No
If user ask to continute
Delete this break point from break point table

Process Flow Diagram of Decode:

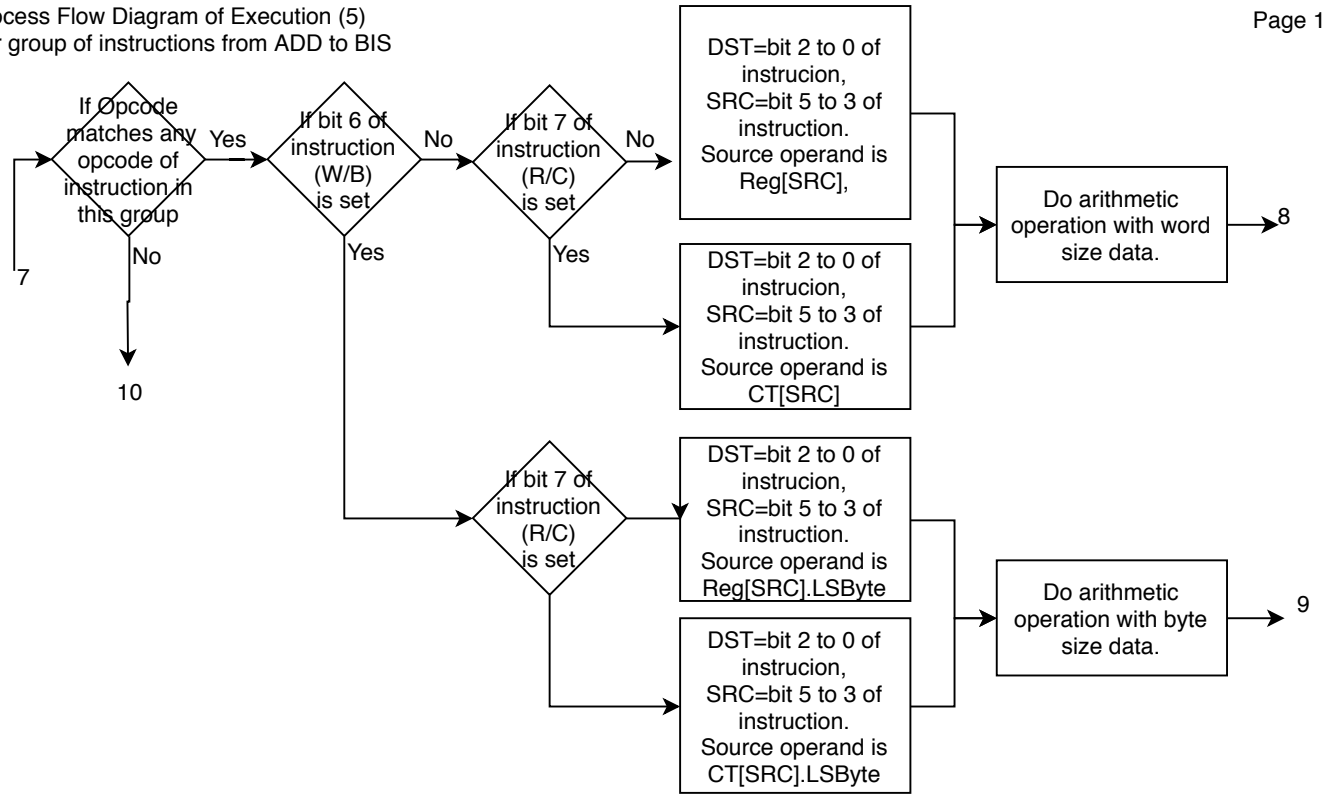


Process Flow Diagram of Execution (1)

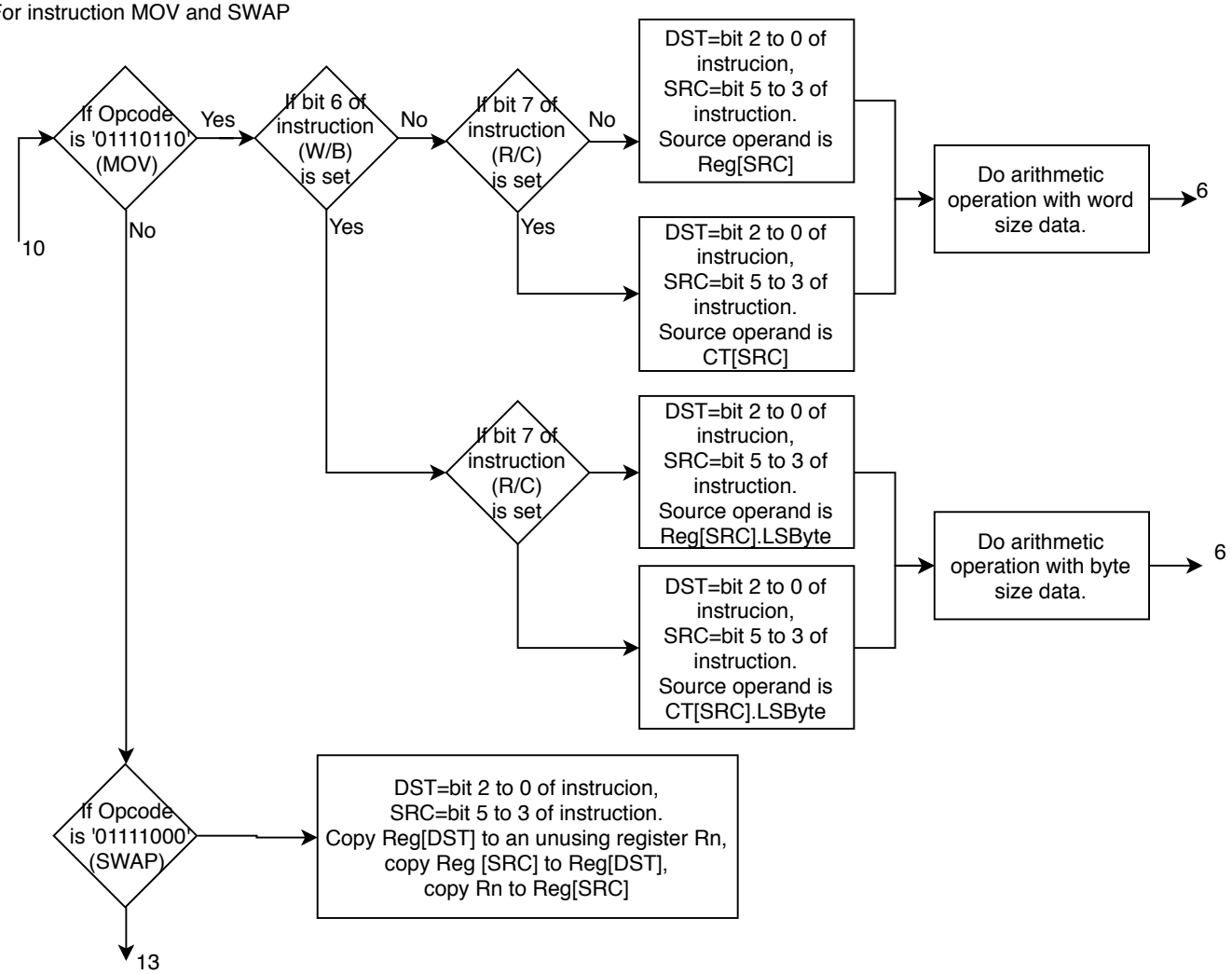


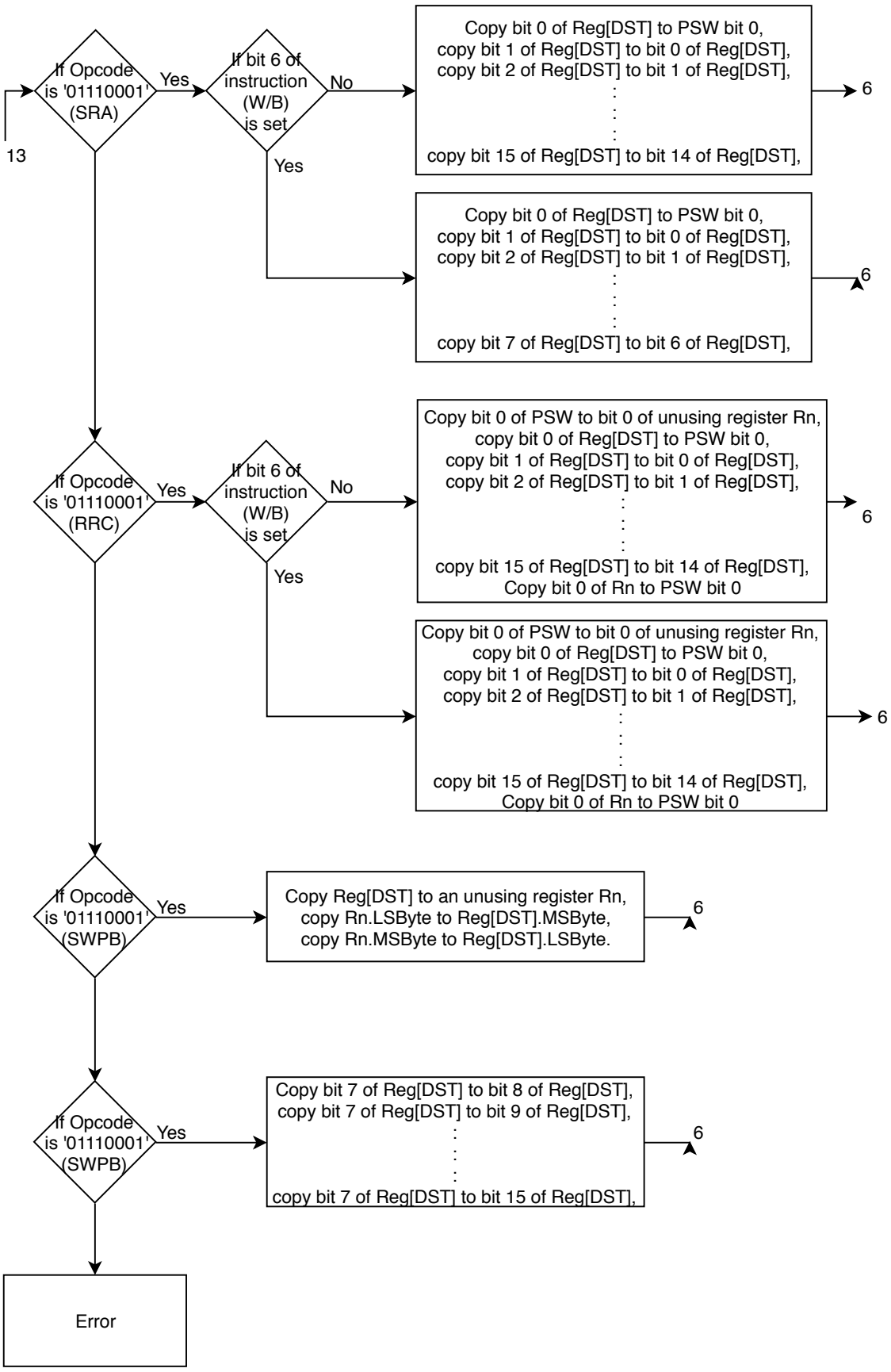

4
If Opcode is '110' (LDR)
Yes
No
Set offset bit 6 to 1 = bit 10 to 3 of instruction, offset bit 0 = 0, fill up offset bit 15 to 7 with bit 10 of instruction, DST=bit 2 to 0 of instrucion, SRC=bit 5 to 3 of instruction, EA = Reg[SRC]+offset
If bit 6 of instruction (W/B) is set
No
Yes
Reg[DST].HSByte = mem[EA], Reg[DST].LSByte = mem[EA+1]
Reg[DST].LSByte = mem[EA]
If Opcode is '111' (STR)
Set offset bit 6 to 1 = bit 10 to 3 of instruction, offset bit 0 = 0, fill up offset bit 15 to 7 with bit 10 of instruction, DST=bit 2 to 0 of instrucion,SRC=bit 5 to 3 of instruction, EA = Reg[SRC]+offset
If bit 6 of instruction (W/B) is set
No
Yes
mem[EA] = Reg[SRC].HSByte, mem[EA+1] = Reg[SRC].LSByte
mem[EA] = Reg[SRC].LSByte
If Opcode is '10010' (MOVL)
DST=bit 2 to 0 of instrucion, Set bit 7 to 0 of Reg[DST] = bit 10 to 3 of instruction
If Opcode is '10011' (MOVLZ)
DST=bit 2 to 0 of instrucion,Set bit 7 to 0 of Reg[DST] = bit 10 to 3 of instruction, set bit 15 to 8 of Reg[DST] = 0000 0000
If Opcode is '10100' (MOVH)
DST=bit 2 to 0 of instrucion,Set bit 15 to 8 of Reg[DST] = bit 10 to 3 of instruction
If Opcode is '000' (BL)
Offset bit 12 to 0 = bit 12 to 0 of instruction, fill up offset bit 15 to 13 with bit 12 of instruction, set LR = PC, and set PC = PC+offset
Finish Execution
5

Process Flow Diagram of Execution (5)
For group of instructions from ADD to BIS

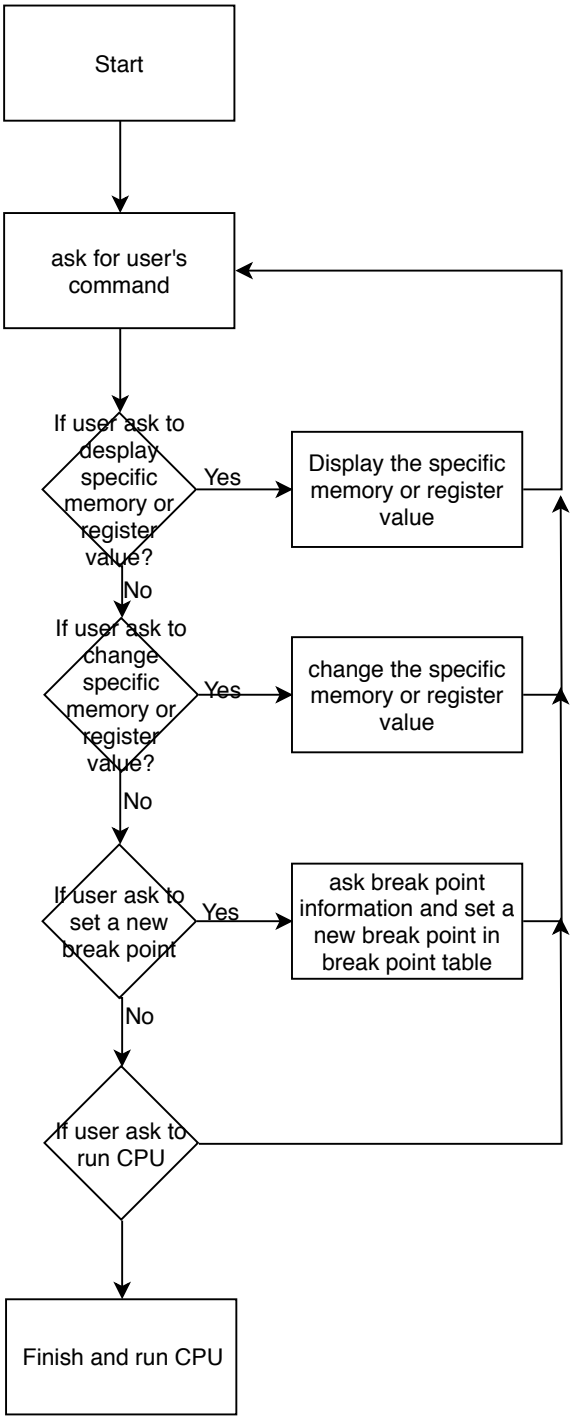
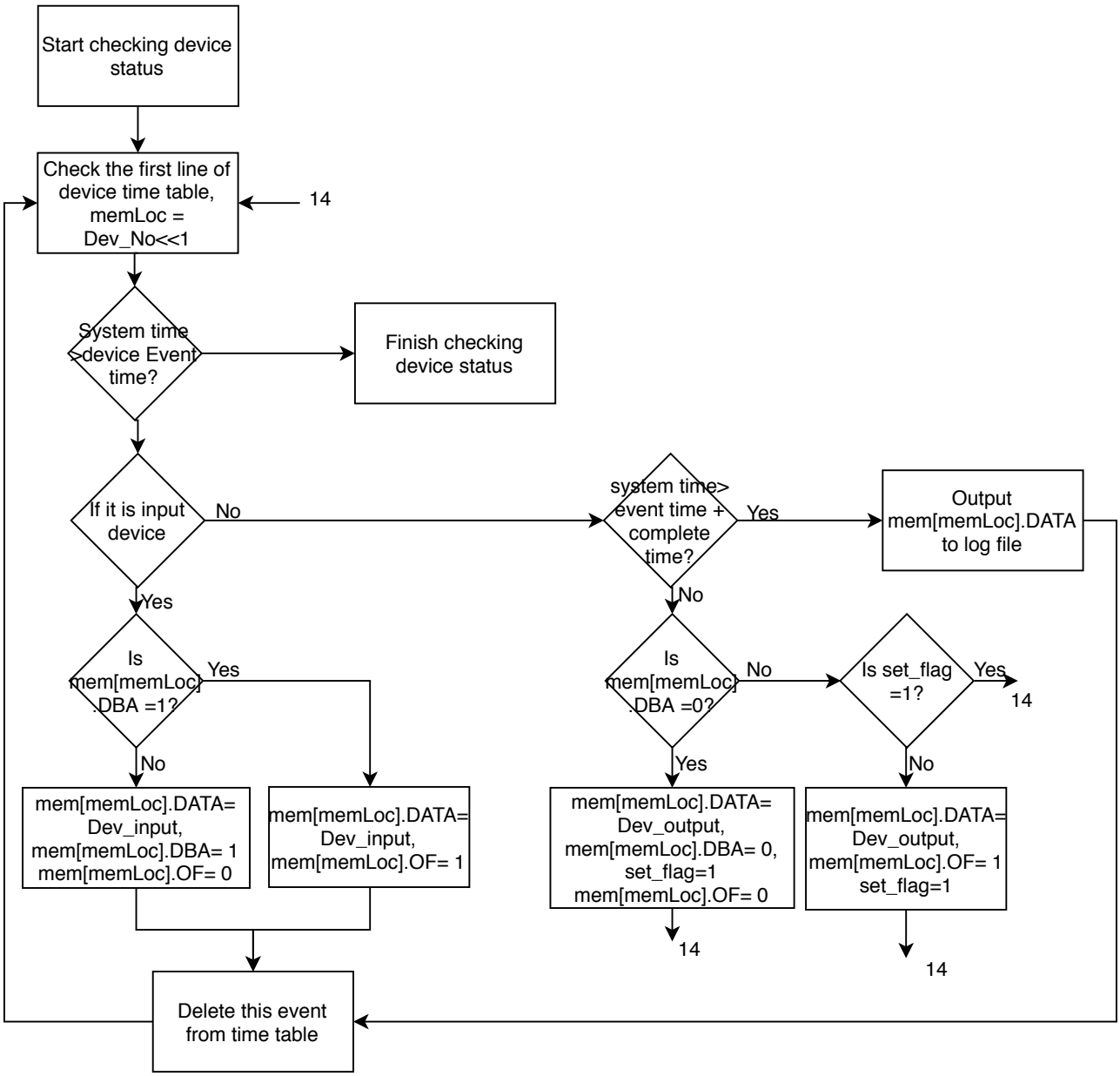


Process Flow Diagram of Execution (6)
For instruction MOV and SWAP




13
If Opcode is '01110001' (SRA)
Yes
If bit 6 of instruction (W/B) is set
No
Yes
Copy bit 0 of Reg[DST] to PSW bit 0,
copy bit 1 of Reg[DST] to bit 0 of Reg[DST],
copy bit 2 of Reg[DST] to bit 1 of Reg[DST],
copy bit 15 of Reg[DST] to bit 14 of Reg[DST],
6
Copy bit 0 of Reg[DST] to PSW bit 0,
copy bit 1 of Reg[DST] to bit 0 of Reg[DST],
copy bit 2 of Reg[DST] to bit 1 of Reg[DST],
copy bit 7 of Reg[DST] to bit 6 of Reg[DST],
6
If Opcode is '01110001' (RRC)
Yes
If bit 6 of instruction (W/B) is set
No
Yes
Copy bit 0 of PSW to bit 0 of unusing register Rn,
copy bit 0 of Reg[DST] to PSW bit 0,
copy bit 1 of Reg[DST] to bit 0 of Reg[DST],
copy bit 2 of Reg[DST] to bit 1 of Reg[DST],
copy bit 15 of Reg[DST] to bit 14 of Reg[DST],
Copy bit 0 of Rn to PSW bit 0
6
Copy bit 0 of PSW to bit 0 of unusing register Rn,
copy bit 0 of Reg[DST] to PSW bit 0,
copy bit 1 of Reg[DST] to bit 0 of Reg[DST],
copy bit 2 of Reg[DST] to bit 1 of Reg[DST],
copy bit 15 of Reg[DST] to bit 14 of Reg[DST],
Copy bit 0 of Rn to PSW bit 0
6
If Opcode is '01110001' (SWPB)
Yes
Copy Reg[DST] to an unusing register Rn,
copy Rn.LSByte to Reg[DST].MSByte,
copy Rn.MSByte to Reg[DST].LSByte.
6
If Opcode is '01110001' (SWPB)
Yes
Copy bit 7 of Reg[DST] to bit 8 of Reg[DST],
copy bit 7 of Reg[DST] to bit 9 of Reg[DST],
copy bit 7 of Reg[DST] to bit 15 of Reg[DST],
6
Error

Process Flow Diagram for
run debugger



Process flow diagram for checking device status