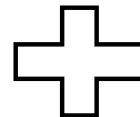
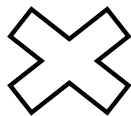


### Problem 3: Return of Tomasulo's Algorithm (30 pts)

The diagram below shows a snapshot at a particular point in time of various parts (reservation stations and register alias table) of the microarchitecture for an implementation supporting out-of-order execution in the spirit of Tomasulo's Algorithm. Note that there is an adder and a multiplier in this machine. The processor is supplied with a seven instruction program following reset. The state below was captured at some point in time during the execution of these seven instructions. Anything marked with a – is unknown and can't be relied upon for your answer. You should assume that the bottommost instruction in the reservation station arrived earliest and the topmost instruction in the reservation station arrived last.

SRC 1				SRC 2		
ID	V	Tag	Val	V	Tag	Val
-	-	-	-	-	-	-
E	0	C	-	0	A	-
F	0	A	-	1	-	20
C	0	A	-	0	A	-

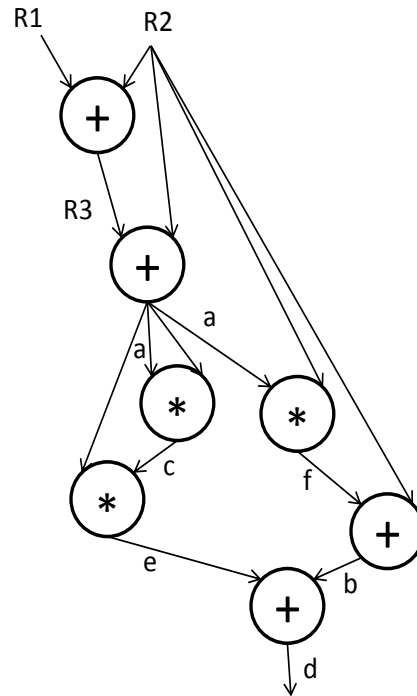
SRC 1				SRC 2		
ID	V	Tag	Val	V	Tag	Val
-	-	-	-	-	-	-
D	0	B	-	0	E	-
B	1	-	20	0	F	-
A	1	-	20	1	-	30



RAT			
Reg	V	Tag	Val
R0	1	-	5
R1	0	A	10
R2	0	B	20
R3	1	-	30
R4	0	C	40
R5	0	D	50
R6	1	-	60
R7	1	-	70

(Question 3 cont'd)

**A) [15 pts]** Identify the instructions and draw the data flow graph for the seven instructions (use + for ADD and \* for MUL). Please label the edges of the data flow graph with the destination register tag if known. Label with register number if the tag is not known. Note that the first instruction is an ADD with destination register R3.



**B) [15 pts]** Fill in the instruction opcodes, source, and destination registers in the table below.

Instructions			
OP	DEST	SRC1	SRC2
ADD	R3	R1	R2
ADD	R1	R2	R3
MUL	R4	R1	R1
MUL	R5	R2	R1
ADD	R2	R2	R5
MUL	R5	R4	R1
ADD	R5	R2	R5