1. The RTN for a new SRC instruction is given by: INC (:= op = 30) → R[ra] ← R[rc] + 1:
   a. Give the Hex. representation of the SRC assembly language instruction: INC R26, R17 (10 pts.)
   b. Give the concrete RTN and the control sequence required to execute the instruction given in part a above on the architecture shown in Figure 1. (10 pts.)

a. 30 26 0 17 = 11110 11010 00000 10001 = F6811 since an instruction is 32 bits = F6811000
op ra rb rc

b. W ← R17 + 1 -- $R17_{out}$, W_{in} ; Anything on the bus is incremented by 1 and the result is available at the
; input to the W register
R26 ← W -- W_{out}, $R26_{in}$; The result is placed in R[ra]

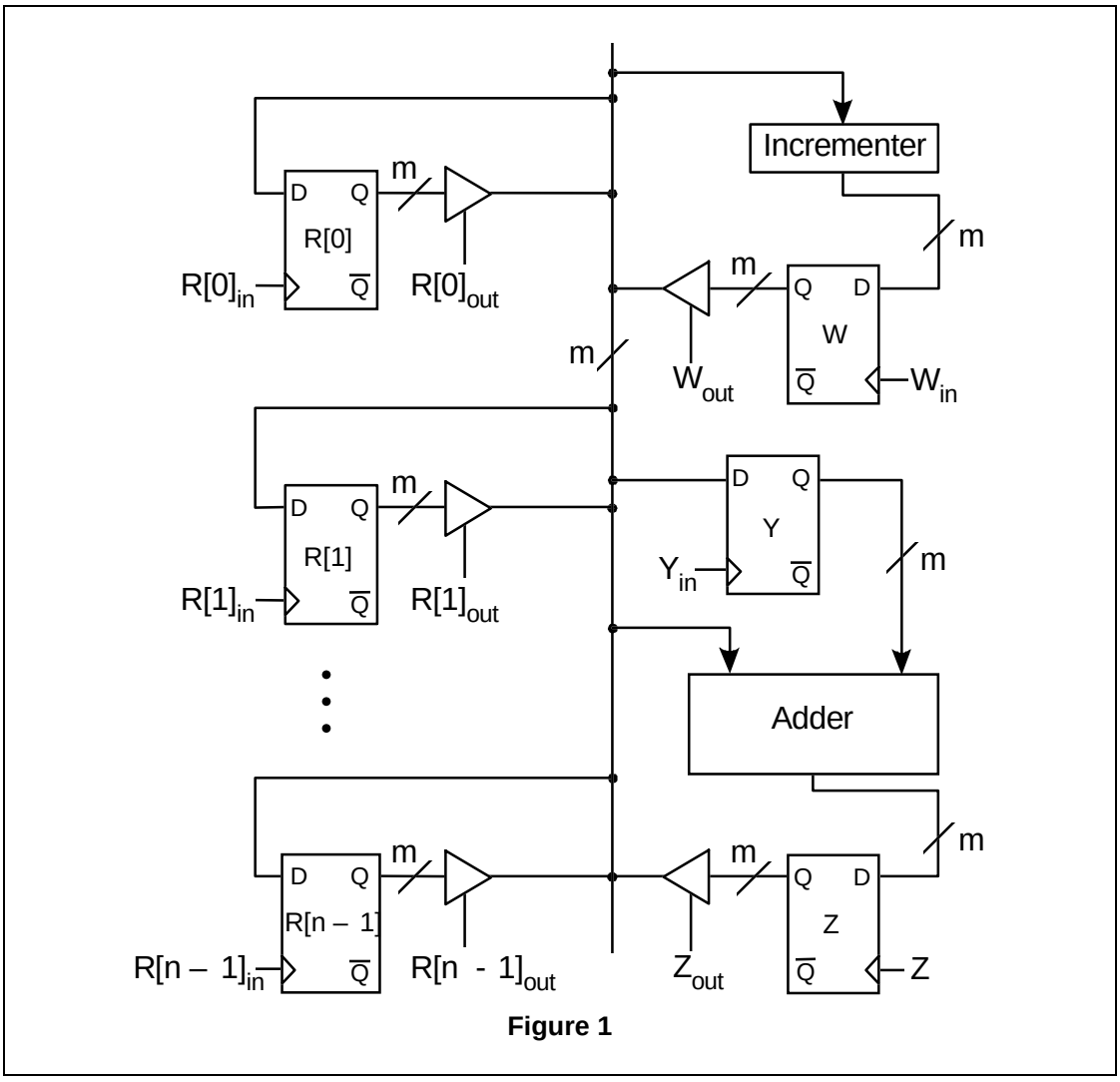
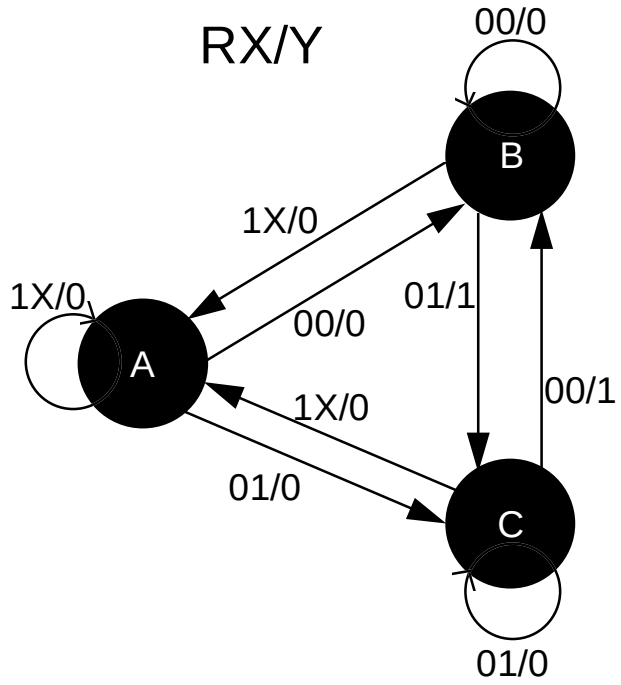


Figure 1

2. A finite state machine (FSM) has 2 inputs, X and R, and a single output Y. When R is true it resets the machine to its initial state and sets the output to 0. When R is 0, the machine outputs a 1 only when the last value of X differs from the previous one. Draw the transition diagram for the FSM. (15 pts.)



3. Shown below is a SRC assembly language program that is executed.

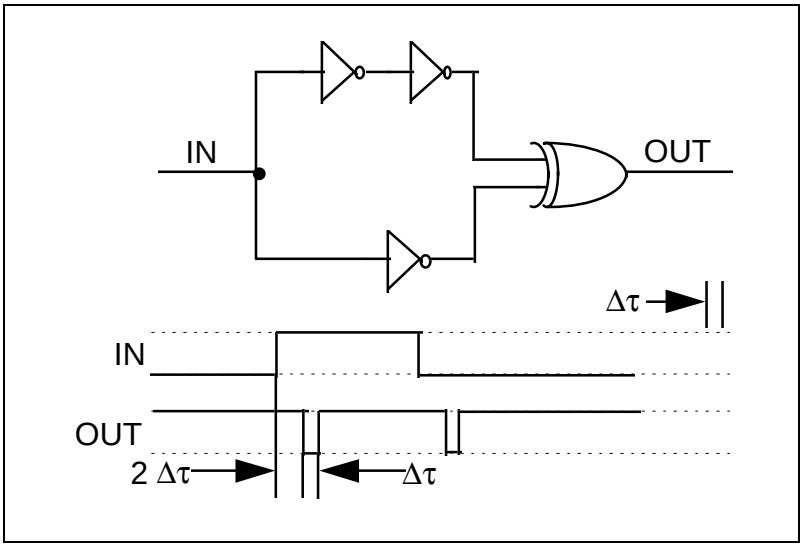
   a. ***On the line next to each instruction that is actually executed***, show the contents of all registers that are changed as a result of ***fetching and executing*** the instruction. ***If the instruction is not fetched and executed, write "NFE".*** (15 pts.)
   b. Show the Hex. values that are generated by the assembler for the lines of source code that are underlined above. See the table of Op-Codes at the end of the exam. (10 pts.)

		Location	
	.org 1000H		
endtest:	.dc -12345		
	.org 5000H		
	ldr r10, endtest	5000H	PC ← 5004H, r10 ← -12345
	lar r30, start	5004H	PC ← 5008H, r30 ← 500CH
	lar r20, end	5008H	PC ← 500CH, r20 ← 5018H
start:	brlmi r5, r20, r10	500CH	PC ← 5018H, r5 ← 5010H
	neg r10, r10	5010H	NFE
	br r30	5014H	NFE
end:	stop	5018H	PC ← 501CH, (Run ← 0)

The op-code for the ldr is 2 and the register is r10 giving 00010 01010 The remaining 22 bits contain the displacement from the current value of the PC, 5004H, to "endtest" = 1000H. The difference is -4004H. To get the negative value form the 2's C and extend to 22 bits. 0100 0000 0000 0100 → 1011 1111 1111 1100. The complete instruction is therefore: 00010 01010 11 1111 1011 1111 1111 1100 = 12BFBFFCH.

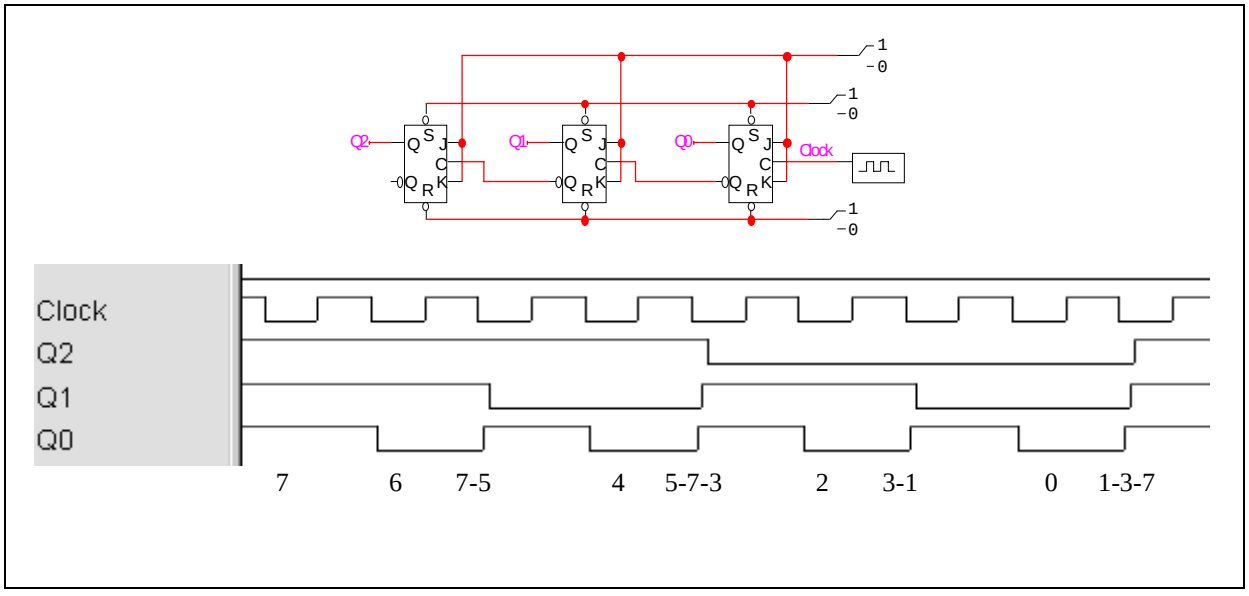
The opcode for the brl is 9 and the 3 register fields contain 5, 20, and 10 giving 01001 00101 10100 01010. The low order 3 bits of the instruction are 101 indicating a branch when register 10 contains a negative number. This gives 01001 00101 10100 01010 0000 0000 0101 = 4968A005H.

4. Shown below is a circuit and its input waveform. On the diagram, draw the output. A gate delay is shown on the diagram. Label the output to indicate the number of gate delays from the change in the input that caused the change in the output. (15 pts.)



5 a. Use flip-flops and any gates that are necessary to draw a circuit for a 3 bit ripple down counter that will count from 7 to 0. You need not show the gates that make up the flip-flops but you must indicate the type of flip-flops used and label all inputs and outputs. (10 pts.)

b. Draw the timing diagrams for the clock and the Q outputs taking propagation delays into account. Under the timing diagrams show the decimal values of the count for each transition in the output. (15 pts.)



0 = nop 1 = ld 2 = ldr 3 = st 4 = str 5 = la 6 = lar 8 = br
9 = brl 12 = add 13 = addi 14 = sub 15 = neg 20 = and 21 = andi 22 = or
23 = ori 24 = not 26 = shr 27 = shra 28 = shl 29 = shc 31 = stop