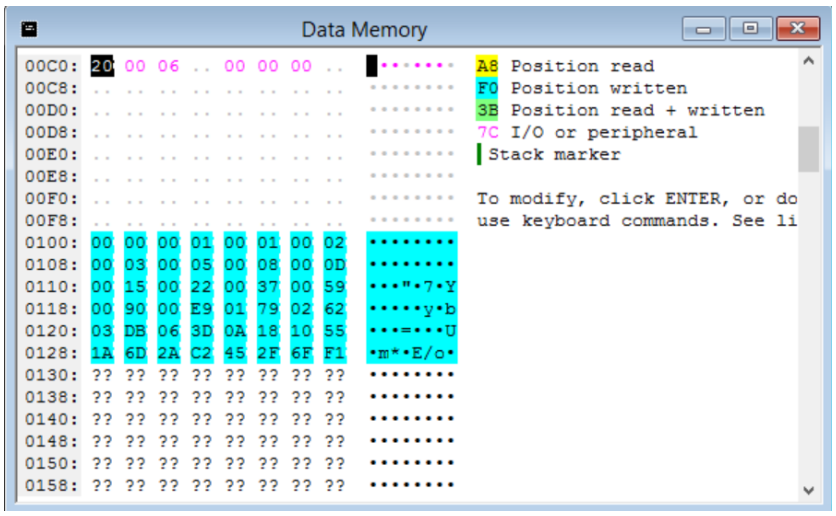
Philip Nevins
5/9/2022
ENGR 271

HW#4, Problem #1

1. There is a sequence of numbers called the Fibonacci sequence: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144,… (see: https://oeis.org/A000045). Using VMLAB write a short AVR program to calculate and move the numbers into SRAM memory starting at address 0x100. Use two bytes per number. The fourth number is 2, so the memory address 0x106 should have the byte 0x00 and address 0x107 should have the byte 0x02 (big-endian). Stop when the numbers no longer fit in two bytes.

```
.include "C:\VMLAB\include\m168def.inc"
ldi r26, 0
ldi r27, 1
ldi r16, 0
ldi r17, 0
ldi r18, 0
ldi r19, 1                ;initialize values
loop1:                    ;loop to do fib calculations
   st x+, r16
   st x+, r17
   add r17, r19
   adc r16, r18
   st x+, r18
   st x+, r19
   add r19, r17
   adc r18, r16
brcc loop1
```



ENGR 271 | HW#4 Problem#2 | Philip Nevins

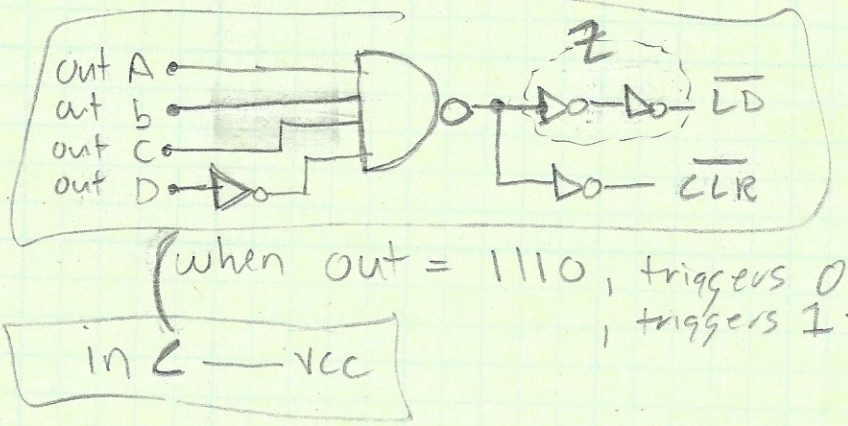
Given: Use a 74163 IC & any other combinational logic required to design a counter circuit that counts the following 11 states. 4, 5, 6 ... 13, 14, 4

Find: Design counter [modulo-11]

Solution:

Modulo-11

	8	2^3	2^2	2^1
(4)	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
⋮				
14	1	1	1	0



From datasheet, when $\overline{LD} = 0$ & $\overline{CLR} = 1$, it's the preset data mode, & it loads ABCD. Since 4 (0100) is lowest #, C → Vcc.

This also matches with doing the math to calculate modulo-11 => (output) − (Vcc input ties)

$15 - 4 = 11$ ✓

We can also put 2 inverters (Z) to ensure $\overline{CLR}$ triggers first, then the 1s value is loaded.

ENGR 271 | HW#4 Problem#3 | Philip Nevins

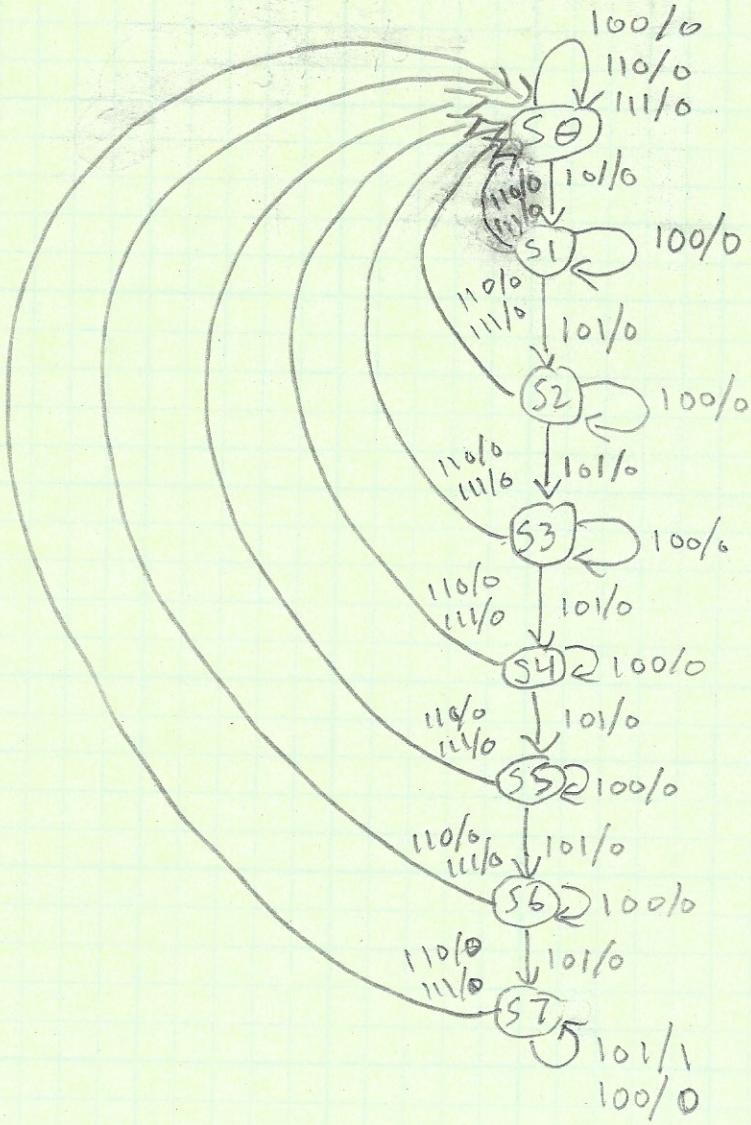
Given: create an unambiguous state diagram for "sticky counter" state machine, with S0-S7. Besides clk, inputs → RST, EN and output - DONE. Machine goes to S0 when RST is High. RST 0 AND EN1 → next state. Once it gets to S7, stays unless RST 1. DONE 1 if & only in state S7 & EN1.

Find: State Diagram

Solution:

$\overline{RST} \cdot EN$ = next state
S7 · RST = S0
S7 · EN = DONE

INPUT: RST, EN (CLK) output: DONE values: CLK/RST/EN/DONE



ENGR 271 | HW#4, Problem#4 | Philip Nevins

Given: Consider adjacency diagram w/ odd # states in a ring. If min. # state bits is used, is it possible to assign state bits w/o critical racing signals?

Find:

Solution: Lets consider 3 states, 2 qbits.

We set $S_0 = 00$, $S_1 = 01$ so we have no racing errors. If we set $S_2 = 11$ to have no racing error at $S_1 \rightarrow S_2$, we have one at $S_2(11) \rightarrow S_0(00)$

If we set $S_2 = 10$, we have a racing error at $S_1(01) \rightarrow S_2(10)$

Lets consider 5 states, 3 qbits.

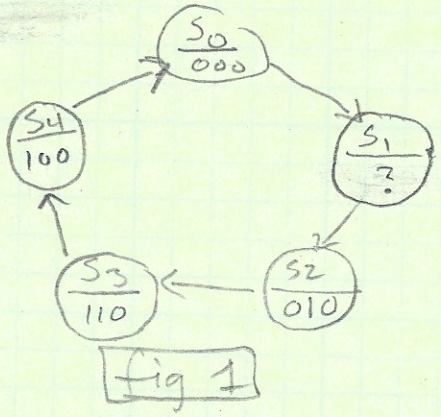


fig 1

We set S_0, S_{2-4} as follows in fig 1. Our only options open are 001, 011, 101, 111.

To eliminate racing error $S_0 \rightarrow S_1$, $S_1 = 001$ only. All other options put a racing error $S_0 \rightarrow S_1$

Now (001) won't work to eliminate racing errors at $S_1(001) \rightarrow S_2(010)$. This instance of having 1 racing error always occurs when we have an odd # of states.

ENGR 271 | HW #4 Problem #5 | Philip Nuins

Given: Design a counter w/ JkFF to produce the following binary sequence: 9, 12, 11, 13, 15, 14, 10, 9...

Find: Design counter (show all design procedure steps)

Solution: Largest # = 15 => 1111 ∴ we need 4 FFs

	Present Q_4	Q_3	Q_2	Q_1	Next Q_4^*	Q_3^*	Q_2^*	Q_1^*	J_4	K_4	J_3	K_3	J_2	K_2	J_1	K_1
9	1	0	0	1	1	1	0	0	X	0	1	X	0	X	X	1
12	1	1	0	0	1	0	1	1	X	0	X	1	1	X	1	X
11	1	0	1	1	1	1	0	1	X	0	1	X	X	1	X	0
13	1	1	0	1	1	1	1	1	X	0	X	0	1	X	X	0
15	1	1	1	1	1	1	1	0	X	0	X	0	X	0	X	1
14	1	1	1	0	1	0	1	0	X	0	X	1	X	0	0	X
10	1	0	1	0	1	0	0	1	X	0	0	X	X	1	1	X

P→N
1→1 Jx, K0
0→1 J1, kx

P→N
0→0 J0, kx
1→0 Jx, k1 ←

Using Jk FF truth table

$J_4 = X$, $K_4 = 0$

J_3

Q_4Q_3 \ Q_2Q_1	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	X 8	X 13	X 15	X 14
10		1 9	1 11	10

$J_3 = Q_4 Q_1$

K_3

Q_4Q_3 \ Q_2Q_1	00	01	11	10
00				
01				
11	1			1
10		X	X	X

$K_3 = Q_4 Q_3 \overline{Q_1}$

J_2

Q_4Q_3 \ Q_2Q_1	00	01	10	10
00				
01				
11	1	1	X	X
10		X		X

$J_2 = Q_4 Q_3$

K_2

Q_4Q_3 \ Q_2Q_1	00	01	11	10
00				
01				
11	X	X		
10		X	1	1

$K_2 = Q_4 \overline{Q_3} Q_2$

→

J_1

Q_4Q_3 \ Q_2Q_1	00	01	11	10
00				
01				
11	1	x	x	
10		x	x	1

K_1

Q_4Q_3 \ Q_2Q_1	00	01	11	10
00				
01				
11	x		1	x
10		1		x

$$J_1 = Q_4 Q_3 \bar{Q}_2 \bar{Q}_1 + Q_4 \bar{Q}_3 Q_2 \bar{Q}_1$$

$$K_1 = Q_4 \bar{Q}_3 \bar{Q}_2 Q_1 + Q_4 Q_3 Q_2 Q_1$$

$$J_1 = Q_4 \bar{Q}_1 (Q_3 \oplus Q_2)$$

$$K_1 = Q_4 Q_1 (Q_3 \oplus Q_2)$$