EE 109 Final Review

"Final" Jeopardy

Binary Brainteasers	1	2	3	4	5
Programming Picklers	1	2	3	4	5
Logic Functions Uncertainties	1	2	3	4	5
Combinational Conundrums	1	2	3	4	5
Sequential Stumpers	1	2	3	4	5
Computer Queries	1	2	3	4	5

"Final" Jeopardy

Binary Brainteasers	1	2	3	4	5
Programming Picklers	1	2	3	4	5
Logic Functions Uncertainties	1	2	3	4	5
Combinational Conundrums	1	2	3	4	5
Sequential Stumpers	1	2	3	4	5
Computer Queries	1	2	3	4	5
2022	1	2	3	4	5

DESIGN PROBLEMS

State Machine Example

- An old TV remote control and TV only supports 2-digit channels: 00-99. Normally, to change the channel we would have to hit two buttons: 38, first 3 then 8, and as soon as you hit the second button it should change the channel. However, for channels 2-9 the remote should allow you to just enter 1-digit and if another button is not pushed soon afterwards should cause the channel to be changed. Implement a state machine that can indicate when the channel should be changed. Have a single input PUSH and single output CHANGE. If you don't push the second button with in 2 clock periods of the first press it should just change the channel no matter what. Four states: 0PUSH, 1PUSH, WAIT, CHGCHAN.

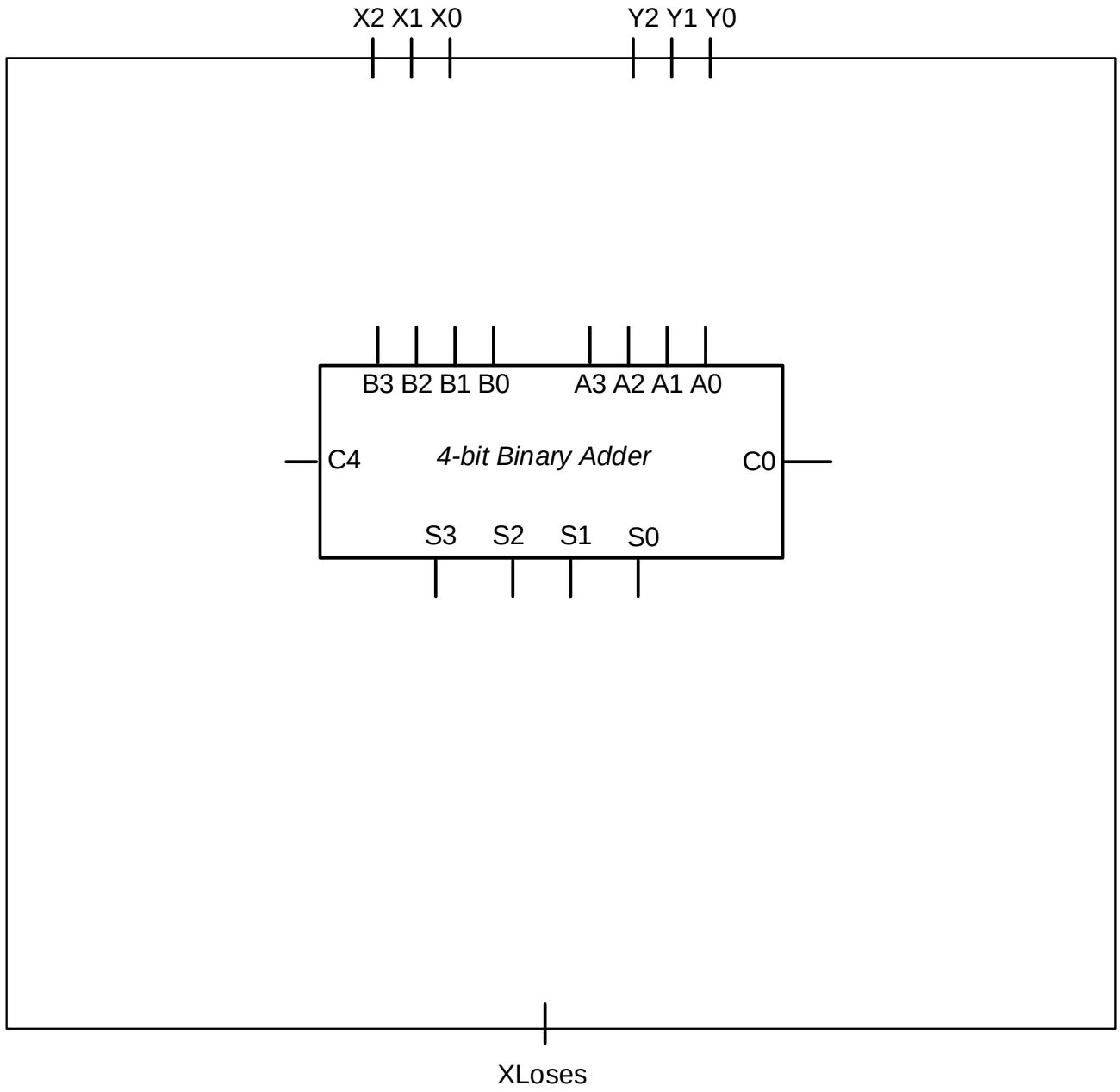
- INPUTS: PUSH
- OUTPUTS: CHANGE

State Machine Example

	P=0	P=1	Change
0Push			
1Push			
Wait			
ChgChan			

Adder and Combinational Design

- Suppose team X and team Y play a game where their scores range 0-7 decimal. Team X is much better than team Y so they give Y a handicap: to truly win, team X must score 5 points more than team Y. Design a circuit that will produce a signal: XLoses given the two 3-bit unsigned input numbers X[2:0] and Y[2:0] representing the scores of each team.


X2 X1 X0
Y2 Y1 Y0
B3 B2 B1 B0
A3 A2 A1 A0
C4
4-bit Binary Adder
C0
S3 S2 S1 S0
XLoses

ISR

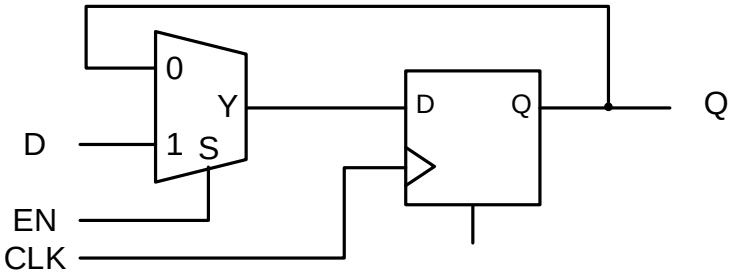
- Determine a prescalar, OCR0A value, and ISR for an 8-bit timer generate at 400Hz square wave for 3 seconds on PD1. Recall the Arduino runs at 16MHz and valid prescalars are: 1, 8, 64, 256, 1024 (choose the smallest prescalar that works).
  - OCR0A: _____
  - Prescalar: _____

ISR Code

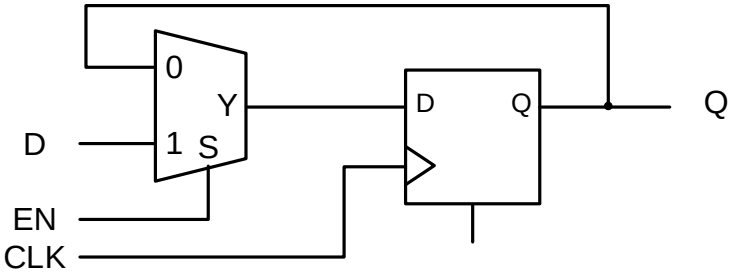
```
ISR(TIMER0_COMPA_vect)
{




}
```

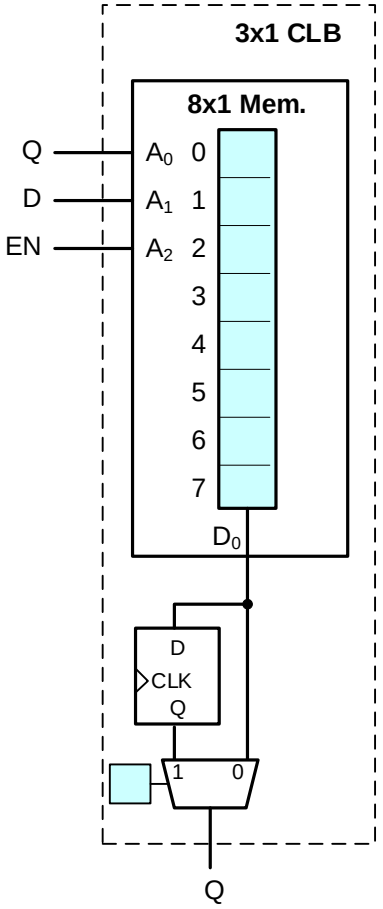
FPGA

- Show how to implement this flip-flop with load enable by determining the values to program into an FPGA's configurable logic block shown to the right.



A2	A1	A0	


0
1
S
Y
D
EN
CLK
D
Q
Q


3x1 CLB
8x1 Mem.
Q
D
EN
A0 0
A1 1
A2 2
3
4
5
6
7
D0
D
CLK
Q
1
0
Q

SOLUTIONS

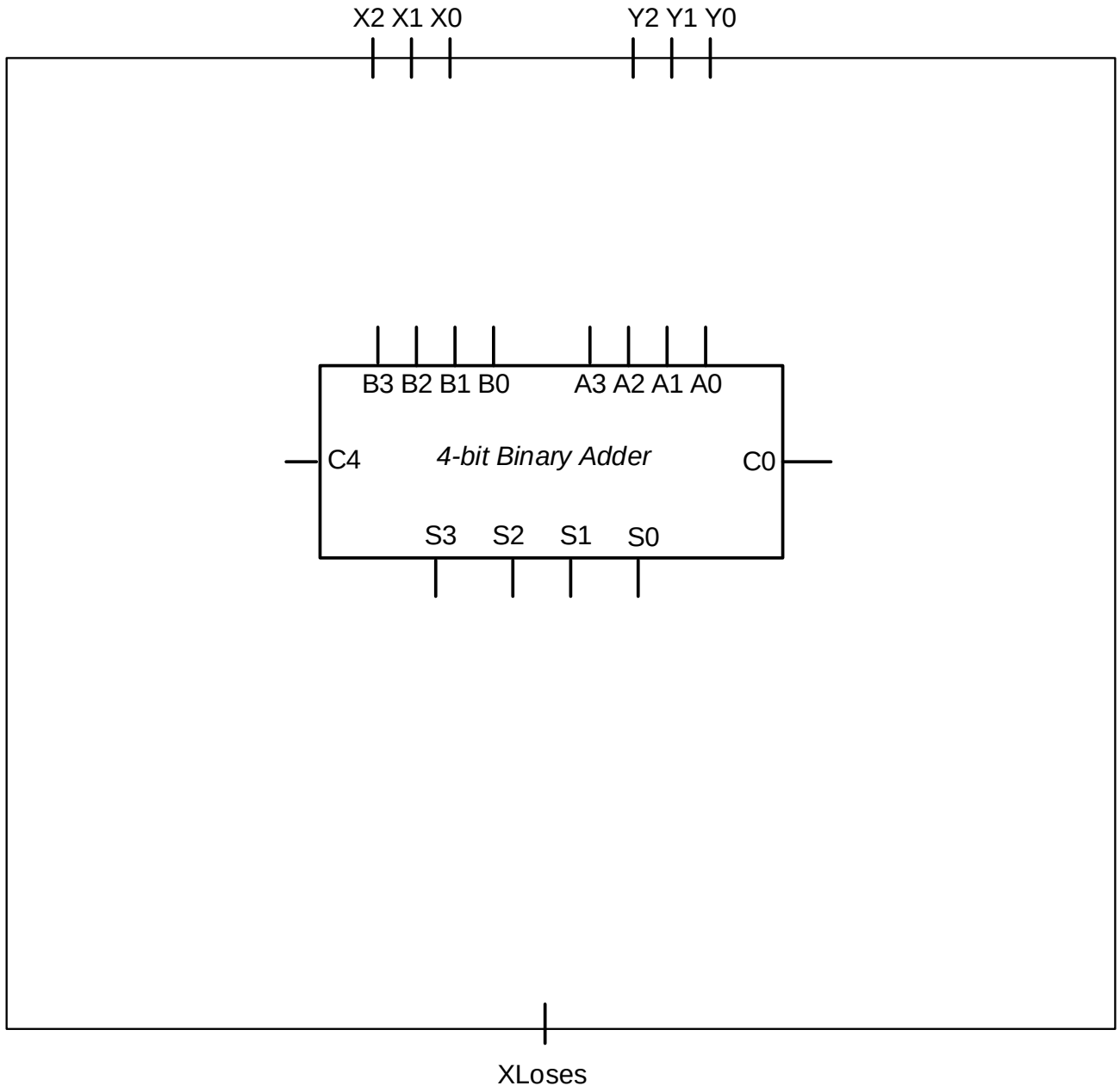
State Machine Example

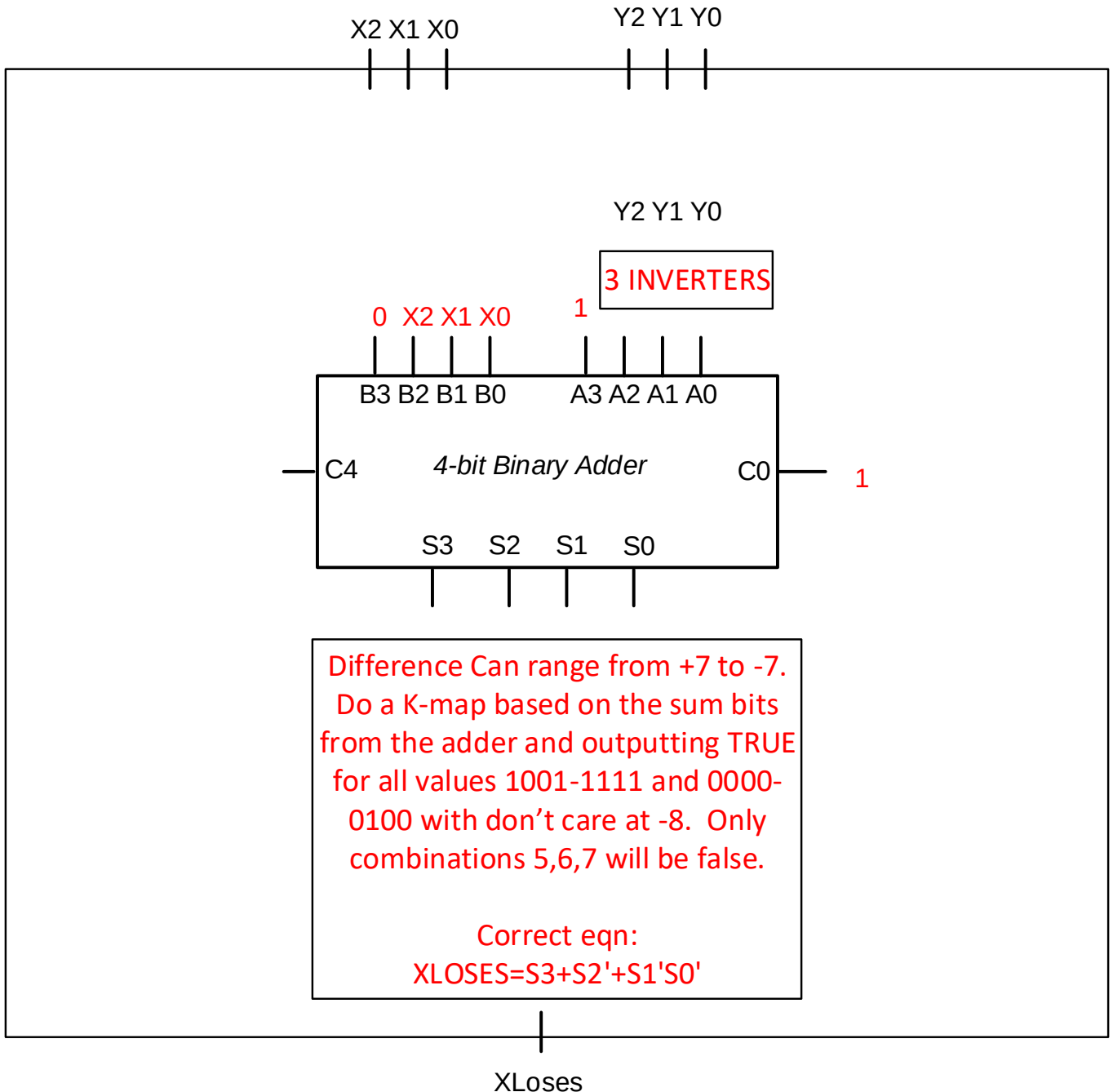
	P=0	P=1	Change
0Push	0Push	1Push	0
1Push	Wait	ChgChan	0
Wait	ChgChan	ChgChan	0
ChgChan	0Push	0Push	1

D0Push = Q0Push*~P + QChgChan
D1Push = Q0Push*P
Dwait = Q1Push*~P
DChgChan = Q1Push*P + Qwait
Change = QChange

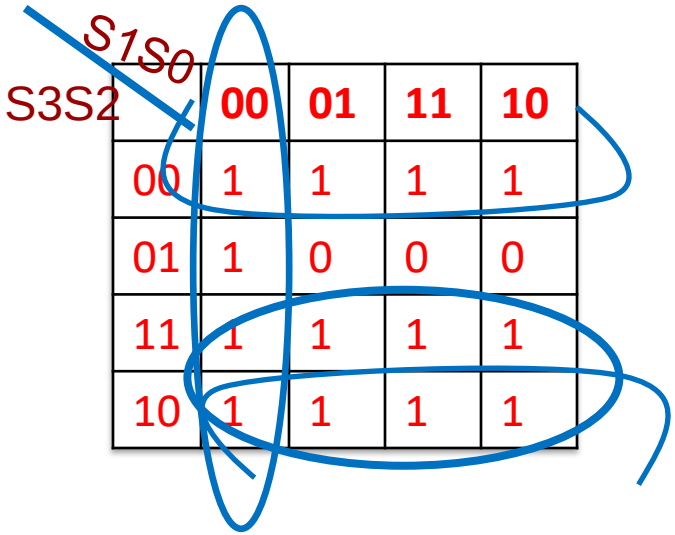
Adder and Combinational Design

- Suppose team X and team Y play a game where their scores range 0-7 decimal. Team X is much better than team Y so they give Y a handicap: to truly win, team X must score 5 points more than team Y. Design a circuit that will produce a signal: XLoses given the two 3-bit input numbers X[2:0] and Y[2:0] representing the scores of each team.


X2 X1 X0
Y2 Y1 Y0
B3 B2 B1 B0
A3 A2 A1 A0
C4
4-bit Binary Adder
C0
S3 S2 S1 S0
XLoses


X2 X1 X0
Y2 Y1 Y0
Y2 Y1 Y0
3 INVERTERS
0 X2 X1 X0
1
B3 B2 B1 B0
A3 A2 A1 A0
C4
4-bit Binary Adder
C0
1
S3 S2 S1 S0
Difference Can range from +7 to -7. Do a K-map based on the sum bits from the adder and outputting TRUE for all values 1001-1111 and 0000-0100 with don't care at -8. Only combinations 5,6,7 will be false.
Correct eqn:
XLOSES=S3+S2'+S1'S0'
XLoses

X-Y	S3S2S1S0	XLOSES
0	0111	1
1	0110	1
2	0101	1
3	0100	1
4	0100	1
5	0101	0
6	0110	0
7	0111	0
-8	1000	1
-7		1
-6		1
-5		1
-4		1
-3		1
-2		1
-1		1



S3+S2'+S1'S0'

ISR

- Determine a prescalar, OCR0A value, and ISR for an 8-bit timer generate at 400Hz square wave for 3 seconds on PD1. Recall the Arduino runs at 16MHz and valid prescalars are: 1, 8, 64, 256, 1024 (choose the smallest prescalar that works).

Timer/ISRs

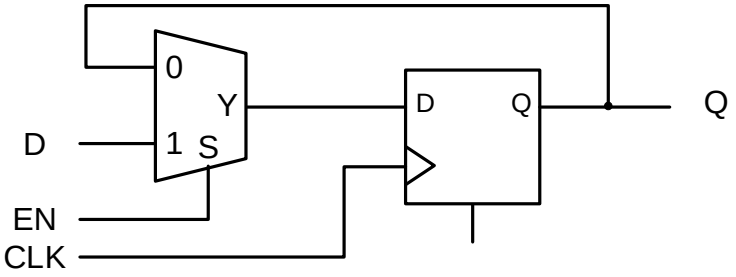
- 400HZ = 1/400 second = 2.5ms
- So ISR at half that rate: 1.25ms
- 800 ISR/sec; 2400 total ISRs for 3 secs
- 16MHz clk * 1.25ms = 20,000 clocks but an 8-bit counter can hold 255 max
  - Prescalar of 64 => 20000/64 = 312.5
  - Prescalar of 256 => 20000/256 = 78.125
  - Prescalar of 1024 => 20000/1024 = 19.53125
- Choose prescalar of 256 and set OCR to 78

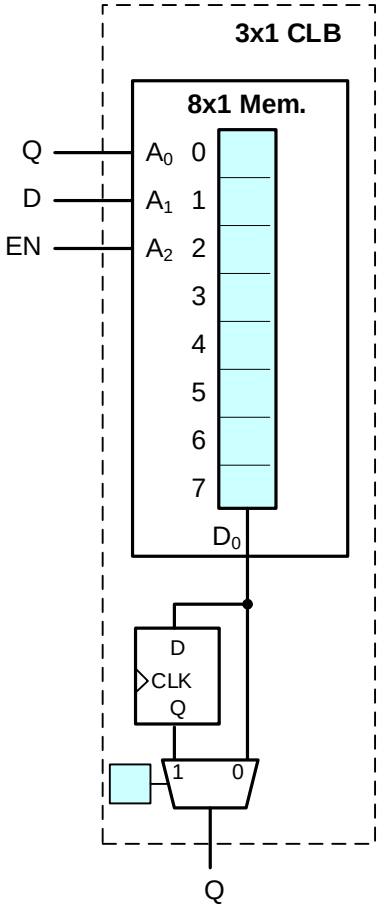
Solutions

```
int cnt=0;
ISR(TIMER0_COMPA_vect)
{
    if (cnt < 2400) {
       PORTD ^= (1<<1);
       cnt++;
   }
   else {
    cnt=0;
    // turn off prescalar – we wouldn't expect
    // you to know the exact bits.
   }
}
```

A2 A1 A0

En	D	Q	Q*
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1





A2 A1 A0

En	D	Q	Q*
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1