

CSE 140 Discussion 9

Speaker: Tim (Wonsuk) Jang

Announcements

- Final HW #4 out, due Wed. 6/1/2022
- Exam 3 out next weekend
 - Can drop lowest exam
- Midterm 2 and HW3 Out Soon

Today's Agenda

- Decoders
- Multiplexers
- RTL Design

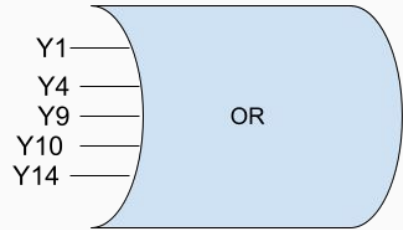
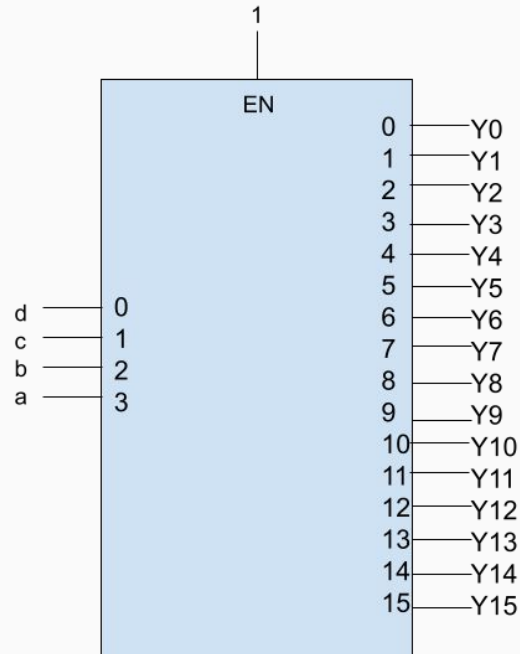
Decoders

$$f(a, b, c, d) = \sum m(1, 4, 9, 10, 14) + \sum d(2, 5, 6, 13, 15)$$

Implement the following function using a minimal number of specified decoders and OR gates.

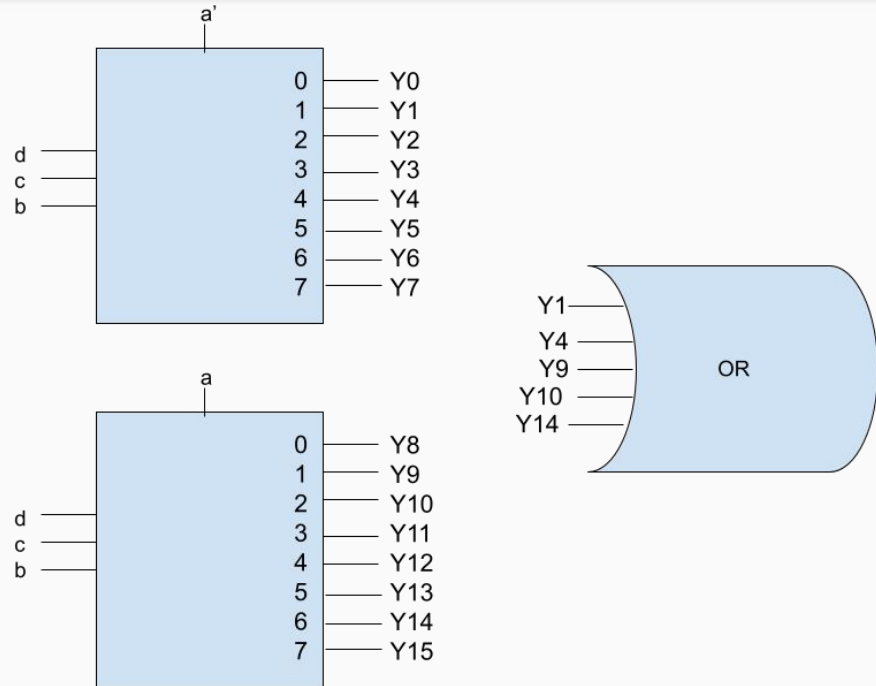
Using 4:16 Decoders and OR gates

- Enumerate the inputs directly
- OR all the minterms
- Always ignore don't cares!



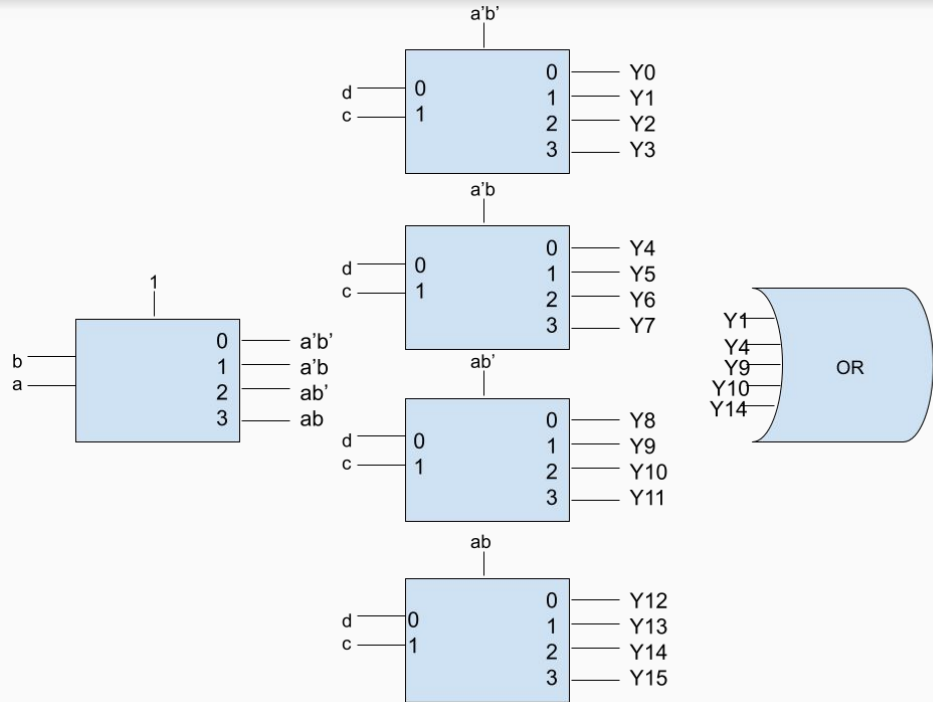
Using 3:8 Decoders and OR gates

- Split on the most significant bit ***a***
- Pass ***a*** as the enable bit
 - Separates decoders to 0-7 and 8-15



Using 2:4 Decoders and OR gates

- Split on ***a*** and ***b***
- Pass decoded results of first decoder to each enable bit for the other 4 decoders
 - Separates decoders to 0-3, 4-7, 8-11, 12-15



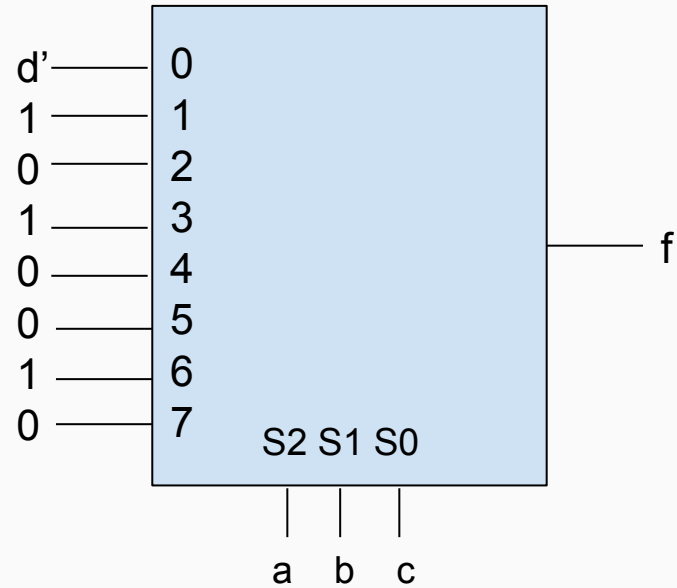
Multiplexers

$$f(a, b, c, d) = \sum m(0, 2, 7, 12, 13) + \sum d(3, 5, 6, 8, 15)$$

- For each $2^n:1$ multiplexer, enumerate first n bits
- Minimize remaining bits and pass in as input to multiplexer
- Pass the enumerated variables as select variables to multiplexer

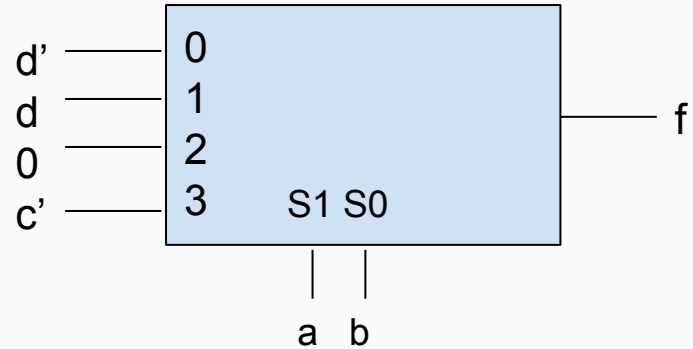
8:1 Multiplexers

a	b	c	d = 0	d = 1	D(d)
0	0	0	1	0	d'
0	0	1	1	X	1
0	1	0	0	X	0
0	1	1	X	1	1
1	0	0	X	0	0
1	0	1	0	0	0
1	1	0	1	1	1
1	1	1	0	X	0



4:1 Multiplexers

a	b	cd=00	cd=01	cd=10	cd=11	D(c,d)
0	0	1	0	1	x	d'
0	1	0	x	x	1	d
1	0	x	0	0	0	0
1	1	1	1	0	x	c'



RTL Design Example

Alg(X,Y,Z,start,U,done)

Input X[7:0], Y[7:0], Z[7:0], start; Output U[7:0], done; Local Object A[7:0], B[7:0], C[7:0]

S1: If start' goto S1;

S2: done <= 0 || A <= X || B <= Y || C <= Z;

S3: A <= Inc(A) || B <= Inc(B);

S4: If A'[7] goto S3 || B <= Inc(B);

S5: If B'[7] goto S4 || C <= Inc(C);

S6: If C'[7] goto S5 || A <= Add(A, B);

S7: U <= C || done <= 1 || goto S1;

End Alg

1. Identify Input and Output of data and control subsystems

Alg(X,Y,Z,start,U,done)

Input **X[7:0], Y[7:0], Z[7:0], start**; Output **U[7:0], done**; Local Object A[7:0], B[7:0],C[7:0]

S1: If start' goto S1;

S2: done <= 0 || A <= X || B <= Y || C <= Z;

S3: A <= Inc(A) || B <= Inc(B);

S4: If A'[7] goto S3 || B <= Inc(B);

S5: If B'[7] goto S4 || C <= Inc(C);

S6: If C'[7] goto S5 || A <= Add(A, B);

S7: U <= C || done <= 1 || goto S1;

End Alg

2. Identify Control Bits to Control Subsystem

Alg(X,Y,Z,start,U,done)

Input X[7:0], Y[7:0], Z[7:0], start; Output U[7:0], done; Local Object A[7:0], B[7:0],C[7:0]

S1: If start' goto S1;

S2: done <= 0 || A <= X || B <= Y || C <= Z;

S3: A <= Inc(A) || B <= Inc(B);

S4: If A'[7] goto S3 || B <= Inc(B);

S5: If B'[7] goto S4 || C <= Inc(C);

S6: If C'[7] goto S5 || A <= Add(A, B);

S7: U <= C || done <= 1 || goto S1;

End Alg

3. Identify Data Subsystem Operations

Alg(X,Y,Z,start,U,done)

Input X[7:0], Y[7:0], Z[7:0], start; Output U[7:0], done; Local Object A[7:0], B[7:0],C[7:0]

S1: If start' goto S1;

S2: done <= 0 || A <= X || B <= Y || C <= Z;

S3: A <= Inc(A) || B <= Inc(B);

S4: If A'[7] goto S3 || B <= Inc(B);

S5: If B'[7] goto S4 || C <= Inc(C);

S6: If C'[7] goto S5 || A <= Add(A, B);

S7: U <= C || done <= 1 || goto S1;

End Alg

4. Map Data Operations to Implementable Functions

A <= X -> A <= Load(X)
B <= Y -> B <= Load(Y)
C <= Z -> C <= Load(Z)
B <= Inc(B) -> B <= INC(B)
C <= Inc(C) -> C <= INC(C)
A <= Add(A,B) -> A <= Add(A,B)
U <= C -> U <= Load(C)

Load Blocks

A <= Load(X)

B <= Load(Y)

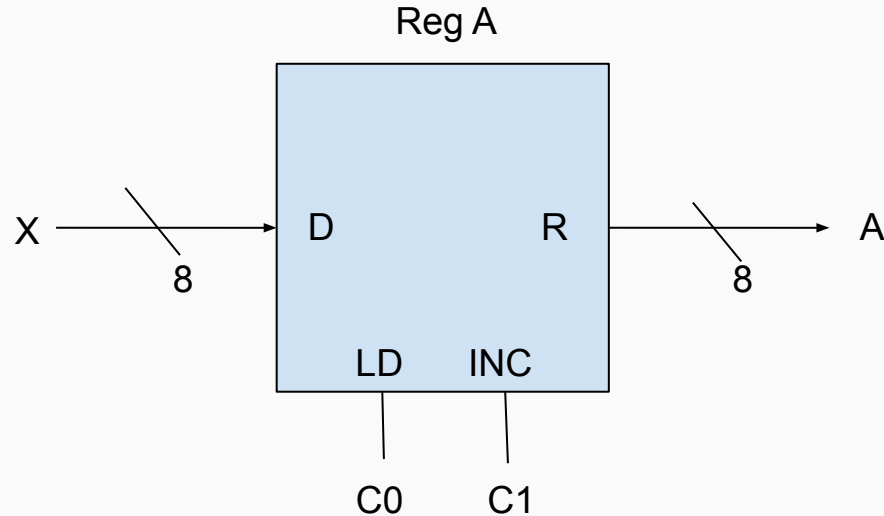
C <= Load(Z)

B <= INC(B)

C <= INC(C)

A <= Add(A,B)

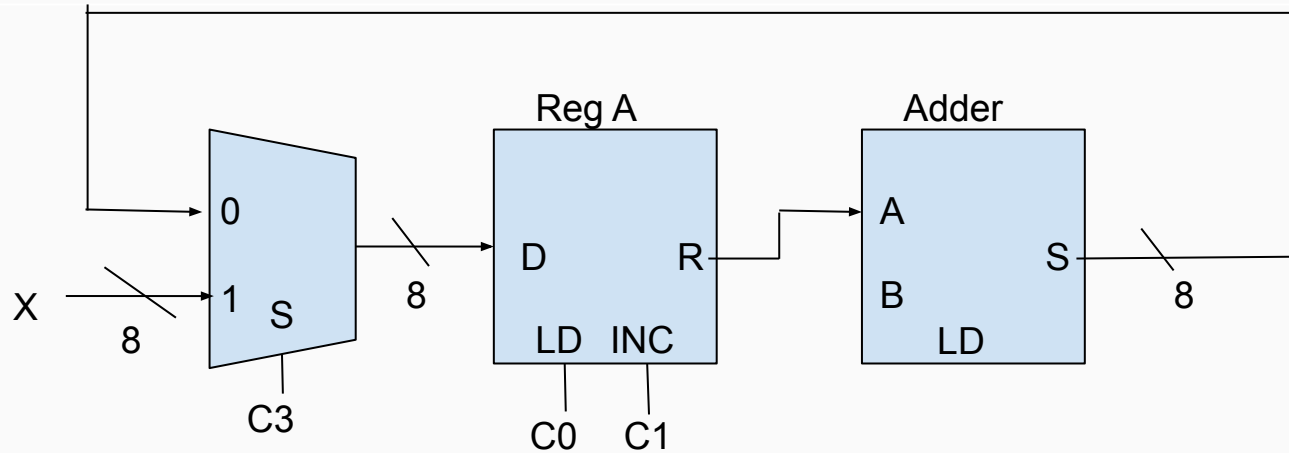
U <= Load(C)



- Reg A, B, and C all have the same format
- Change the inputs, and the load/inc signal (CLR signals can be added if needed)

Blocks with Multiple Assignments

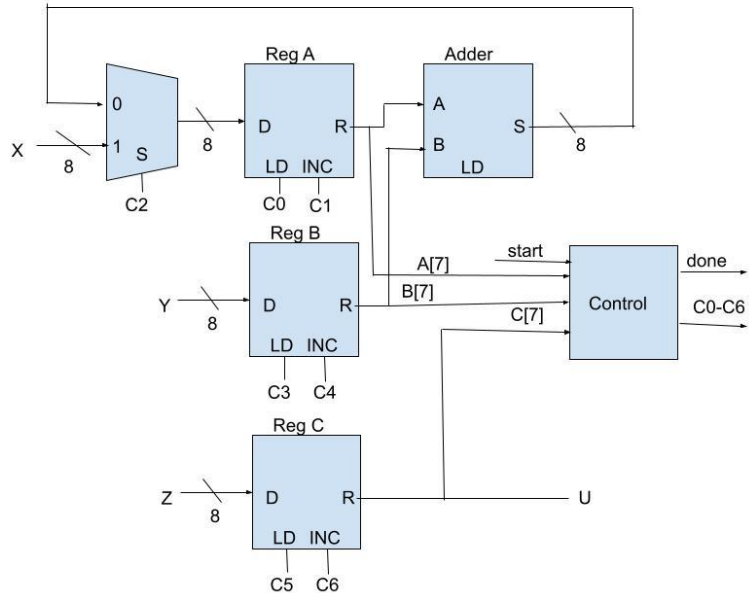
$A \leftarrow \text{Load}(X)$
 $B \leftarrow \text{Load}(Y)$
 $C \leftarrow \text{Load}(Z)$
 $B \leftarrow \text{INC}(B)$
 $C \leftarrow \text{INC}(C)$
 $A \leftarrow \text{Add}(A, B)$
 $U \leftarrow \text{Load}(C)$



- For registers that are assigned multiple values, use Muxes to differentiate each assignment

6. Connect All Systems

- Connect all control signals to the Control Block
- Output final value U
- Control Block outputs the done flag as well as signals C0-C6



Control Subsystem

- Use control signals as transitions to go from each state to another

