Unit5

- Performing operations at different rates
- Using separate state machines running in parallel
  - Find the GCD of different rates
  - e.g. one blinks at 2 Hz, another at 5 Hz, then should stay on/off for 250/100ms, GCD = 50 ms

Unit6

- Step 1: Define a mask that has 1's where the bits are to be copied
  - #define MASKBITS 0xf0
- Step 2: Clear those bits in the destination register using the MASK
  - PORTB &= ~MASKBITS
- Step 3: Shift the bits of x to align them appropriately, then perform the regular step 3
  - PORTB |= ((x<<4) & MASKBITS);

Unit7

Applying Minterms to Synthesize a Function

- Each numbered minterm checks whether the inputs are equal to the corresponding combination. When the inputs are equal, the minterm will evaluate to 1 and thus the whole function will evaluate to 1.

x	y	z	P	use...
0	0	0	0	
0	0	1	0	
0	1	0	1	m_2
0	1	1	1	m_3
1	0	0	0	
1	0	1	1	m_5
1	1	0	0	
1	1	1	1	m_7

$$P = m_2 + m_3 + m_5 + m_7$$
$$= x'yz' + x'yz + xy'z + xyz$$

when x,y,z = {0,1,0} = 2 then

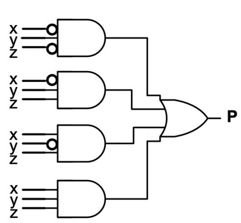
$$P = 0'\bullet 1\bullet 0' + 0'\bullet 1\bullet 0 + 0\bullet 1'\bullet 0 + 0\bullet 1\bullet 0$$
$$= 1 + 0 + 0 + 0 = 1$$

when x,y,z = {1,0,1} = 5 then

$$P = 1'\bullet 0\bullet 1' + 1'\bullet 0\bullet 1 + 1\bullet 0'\bullet 1 + 1\bullet 0\bullet 1$$
$$= 0 + 0 + 1 + 0 = 1$$

when x,y,z = {0,0,1} = 1 then

$$P = 0'\bullet 0\bullet 1' + 0'\bullet 0\bullet 1 + 0\bullet 0'\bullet 1 + 0\bullet 0\bullet 1$$
$$= 0 + 0 + 0 + 0 = 0$$



Applying Maxterms to Synthesize a Function

- Each output that should produce a '0' can be checked-for with an OR gate
  - We refer to that OR-gate checker as a Maxterm of the function (M_i) where i represents the decimal value of the binary combination being checked
- We then AND together the maxterms

x	y	z	P	use...
0	0	0	0	M_0
0	0	1	0	M_1
0	1	0	1	
0	1	1	1	
1	0	0	0	M_4
1	0	1	1	
1	1	0	0	M_6
1	1	1	1	

$$P = M_0 \bullet M_1 \bullet M_4 \bullet M_6$$
$$= (x+y+z) \bullet (x+y+z') \bullet (x'+y+z) \bullet (x'+y'+z)$$

when x,y,z = {0,0,1} = 1 then

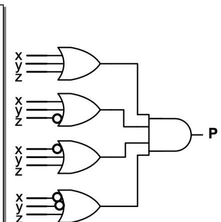
$$P = (0+0+1) \bullet (0+0+1') \bullet (0'+0+1) \bullet (0'+0'+1)$$
$$= 1 \bullet 0 \bullet 1 \bullet 1 = 0$$

when x,y,z = {1,1,0} = 6 then

$$P = (1+1+0) \bullet (1+1+0') \bullet (1'+1+0) \bullet (1'+1'+0)$$
$$= 1 \bullet 1 \bullet 1 \bullet 0 = 0$$

when x,y,z = {1,1,1} = 7 then

$$P = (1+1+1) \bullet (1+1+1') \bullet (1'+1+1) \bullet (1'+1'+1)$$
$$= 1 \bullet 1 \bullet 1 \bullet 1 = 1$$



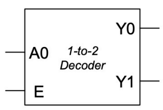
- Using Boolean algebra theorems to simplify
- Applying DeMorgan's theorem and its gate equivalents

Unit8
- K-Maps (up to 4 variables, Don't cares, etc.)
- Implementing an arbitrary combinational function by converting a word description to a truth table and then implementing the circuit using K-Maps

Unit9

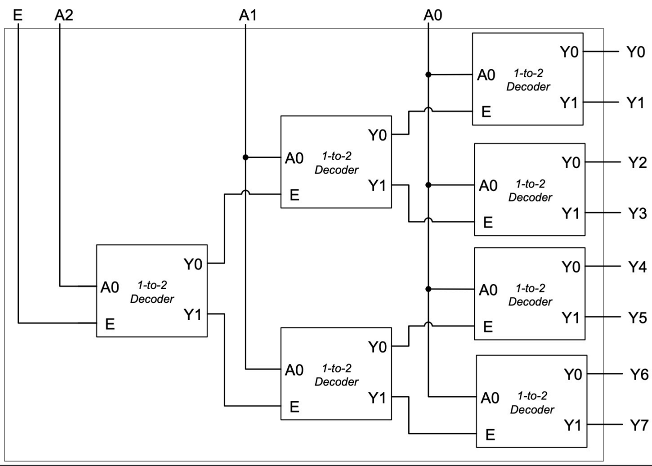
- Build a 3-to-8 decode from 1-to-2 decoders

1-to-2 Decoder Operation



E	A0	Y0	Y1
0	X	0	0
1	0	1	0
1	1	0	1

X = not relevant
(same result for all possible values of A0)



- Full decoders and their implementation
- Enables and decoders
- Mux operation
- Designing muxes at the gate level
- Designing muxes from smaller muxes

Unit10
- Interrupts