

# Principles Of Digital Design

## Discussion: Logic Gates

---

*Subtractor with Simple and Complex Gates*

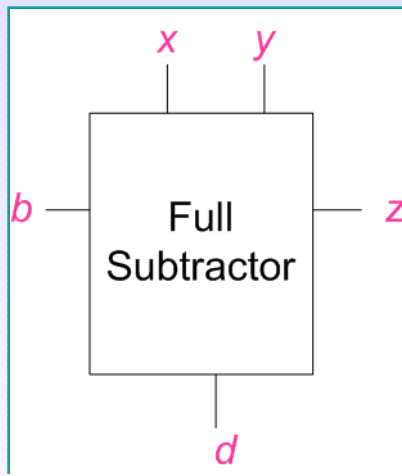
*Low Fuel Detector with Simple and Complex Gates*

# Full Subtractor Design with Simple Gates(1)

- Step 1. Create truth table for full subtractor which has three 1-bit inputs  $x$ ,  $y$  and borrow  $z$ , and two 1-bit outputs difference  $d$  and borrow  $b$  determined by:

- ♦  $d = x - y - z$

- ♦  $b = 1$  if  $x < (y + z)$ , else 0



Subtractor Diagram

<i>A</i>	1	1	1	1	0
<i>B</i>				1	1
<i>Borrows</i>	0	0	1	1	
<i>Difference</i>	1	1	0	1	0

Binary Subtraction Example

$x$	$y$	$z$	$b$	$d$
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

1-bit Subtractor Truth Table

# Full Subtractor Design with Simple Gates(2)

- Step 2. Implement subtractor with simple gates library (Inverter, And, OR, NAND, NOR, XOR, XNOR).

◆ 2-1. Generate Boolean equation using K-map

		xy			
		00	01	11	10
z	0	0	1	0	0
	1	1	1	1	0

$$b = x'(y \oplus z) + yz$$

		xy			
		00	01	11	10
z	0	0	1	0	1
	1	1	0	1	0

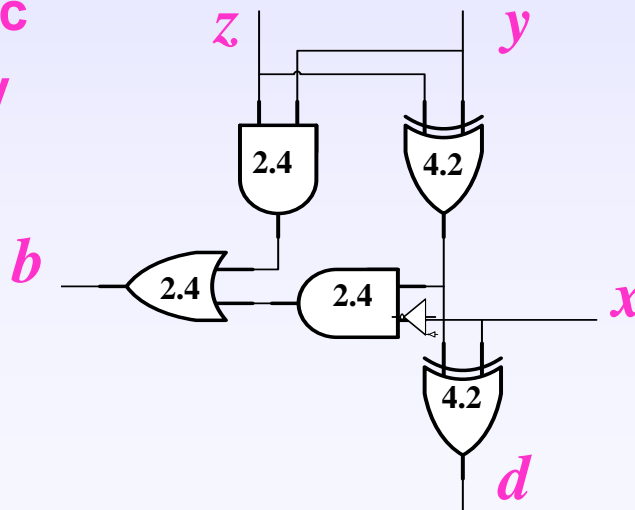
$$d = x'(y \oplus z) + x(y \odot z) = x \oplus y \oplus z$$

◆ 2-2. Draw schematic

◆ 2-3. Calculate delay

z to b = 4.8

x, y to d = 8.4

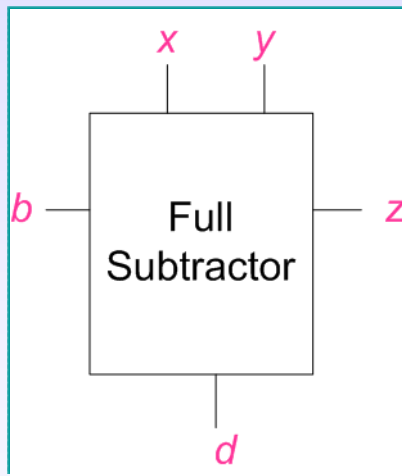


# Full Subtractor Design with Complex Gates(1)

- Step 1. Create truth table for full subtractor which has three 1-bit inputs  $x$ ,  $y$  and borrow  $z$ , and two 1-bit outputs difference  $d$  and borrow  $b$  determined by:

♦  $d = x - y - z$

♦  $b = 1$  if  $x < (y + z)$ , else 0



Subtractor Diagram

<i>A</i>	1	1	1	1	0
<i>B</i>				1	1
<i>Borrows</i>	0	0	1	1	
<i>Difference</i>	1	1	0	1	1

Binary Subtraction Example

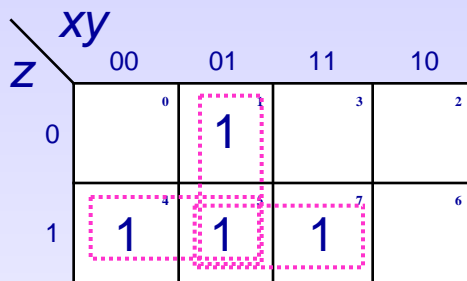
$x$	$y$	$z$	$b$	$d$
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

1-bit Subtractor Truth Table

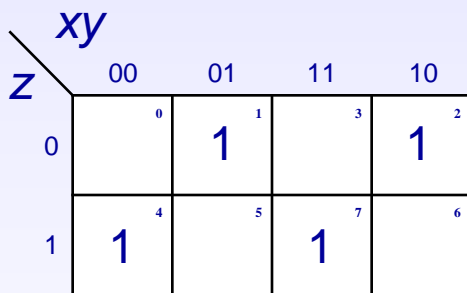
# Full Subtractor Design with Complex Gates(2)

## • Step 2. Implement subtractor with complex gates library

### ◆ 2-1. Generate Boolean equation from K-map



$$b = x'z + x'y + yz$$



$$d = x'y'z + x'yz' + xy'z' + xyz$$

Name	Graphic Symbol	Functional Expression	Delay in ns
2-wide, 2-input AOI		$F = (wx + yz)'$	2.0
3-wide, 2-input AOI		$F = (uv + wx + yz)'$	2.4
2-wide, 3-input AOI		$F = (uvw + xyz)'$	2.2
2-wide, 2-input OAI		$F = ((w + x)(y + z))'$	2.0
3-wide, 2-input OAI		$F = ((u + v)(w + x)(y + z))'$	2.2
2-wide, 3-input OAI		$F = ((u + v + w)(x + y + z))'$	2.4

# Full Subtractor Design with Complex Gates(3)

## ◆ 2-2. Transform Boolean equation to match the gate library

$$\begin{aligned} b &= (x'z + x'y + yz)'' \\ &= ((x'z)' (x'y)' (yz)')' \\ &= ((x+z') (x+y') (y'+z'))' \end{aligned}$$

//De Morgan's Law

//De Morgan's Law

// 3 wide OAI (2.2)

$$d = (x'y'z + x'yz' + xy'z' + xyz)'' \quad //\text{De Morgan's Law}$$

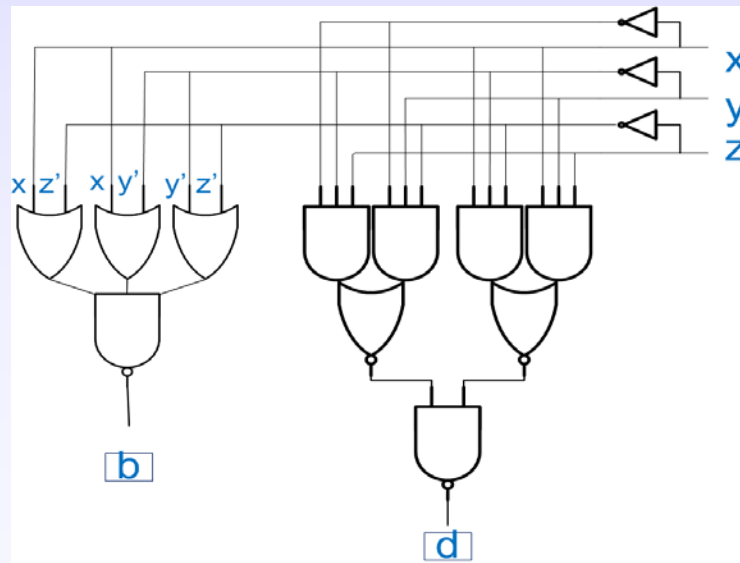
$$= ((x'y'z + x'yz')' (xy'z'+xyz)')' \quad //\text{2 wide AOI (2.2),and NAND(1.4)}$$

## ◆ 2-3. Draw schematic

## ◆ 2-4. Calculate delay

z to b = 3.2

x,y to d = 4.6



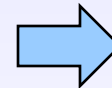
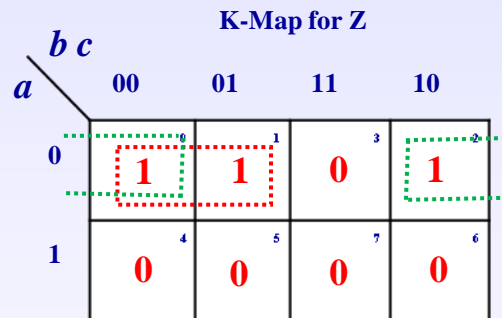
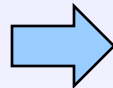
# Low Fuel Detector using Simple Gates

A car has a fuel-level detector that outputs the current fuel-level as a 3-bit binary number, with 000 meaning empty and 111 meaning full. Create a circuit that illuminates a “low fuel” indicator light (by setting an output L to 1) when the fuel level drops below level 3.

## Step 1. Derive a Boolean equation

<i>a</i>	<i>b</i>	<i>c</i>	<i>L</i>
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

Truth Table for Z

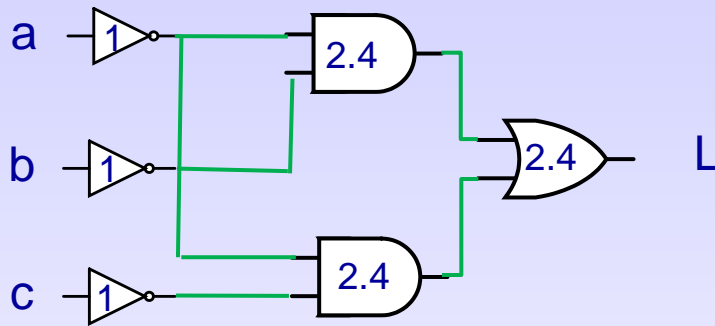


$$L = a'b' + a'c'$$

# Low Fuel Detector using Simple Gates

Step 2. Draw schematic using simple gates

$$L = a'b' + a'c'$$



Step 3. Determine delay

$$\text{delay} = 1 + 2.4 + 2.4 = 5.8$$

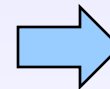
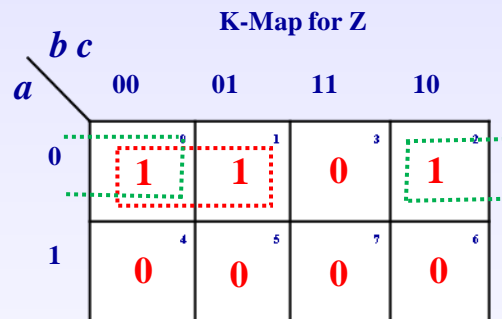
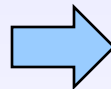
# Low Fuel Detector using Complex Gates

A car has a fuel-level detector that outputs the current fuel-level as a 3-bit binary number, with 000 meaning empty and 111 meaning full. Create a circuit that illuminates a “low fuel” indicator light (by setting an output L to 1) when the fuel level drops below level 3.

## Step 1. Derive a Boolean equation

<i>a</i>	<i>b</i>	<i>c</i>	<i>L</i>
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

Truth Table for Z



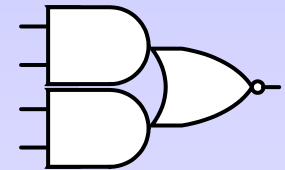
$$L = a'b' + a'c'$$

# Low Fuel Detector using Complex Gates

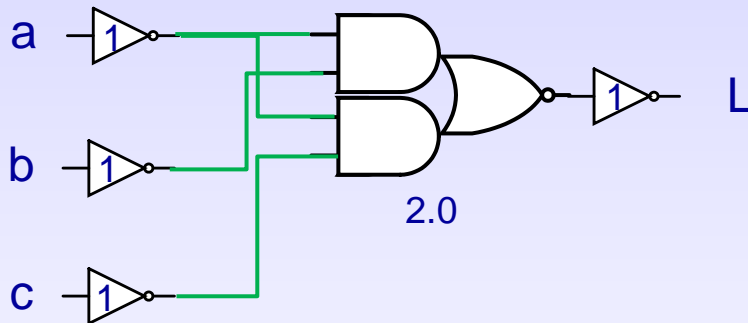
Step 2. Select proper gates from complex gates library

2-wide, 2-input AOI and inverters can be used for equation

$$L = a'b' + a'c'$$



Step 3. Draw schematic using complex gates



Step 4. Determine delay

$$\text{delay} = 1 + 2 + 1 = 4$$