More Efficient Design

- Use already available arithmetic circuit and incorporate logical operations
- Procedure
  - Design the arithmetic section independently
  - Take the circuit, consider $C_{in} = 0$, and determine which logic operations are automatically generated from the arithmetic circuit
  - Modify the circuit to incorporate required but not automatically generated logic operations

More Efficient Design

- Use already available arithmetic circuit and incorporate logical operations

s_1	s_0	Y_i
0	0	0
0	1	B_i
1	0	B_i'
1	1	1

s_1	s_0	X_i	Y_i	Required operation
0	0	A_i	0	OR
0	1	A_i	B_i	XOR
1	0	A_i	B_i'	AND
1	1	A_i	1	NOT

More Efficient Design

- Use already available arithmetic circuit and incorporate logical operations

s_1	s_0	Y_i
0	0	0
0	1	B_i
1	0	B_i'
1	1	1

$S_2 = 0 \rightarrow$ arithmetic

$S_2 = 1 \rightarrow$ logical

Logical part এ $C_{in} = 0$

$A \oplus B$

C_{in} 1 দিলে $A \oplus B \oplus 1$

$=$ not হয়ে যাচ্ছে।

s_2	s_1	s_0	X_i	Y_i	C_i	$F_i = X_i \oplus Y_i$	Operation	Required operation
1	0	0	A_i	0	0	$F_i = A_i$	Transfer A	OR
1	0	1	A_i	B_i	0	$F_i = A_i \oplus B_i$	XOR	XOR
1	1	0	A_i	B_i'	0	$F_i = A_i \odot B_i$	Equivalence (X-NOR)	AND
1	1	1	A_i	1	0	$F_i = A_i'$	NOT	NOT

$x \oplus y \oplus z$

0 দিচ্ছি $= x \oplus y$

$A \oplus \overline{B}$

$= AB + \bar{A}\bar{B}$

Incorporating remaining functions

- Unresolved cases

s_2	s_1	s_0	X_i	Y_i	Automatically Obtained F_i	Required F_i
1	0	0	A_i	0	$F_i = A_i$	$F_i = A_i + B_i$
1	1	0	A_i	B_i'	$F_i = A_i \odot B_i$	$F_i = A_iB_i$

→ AB input দিলে

o/p → $AB \oplus \overline{B}$

$= \overline{AB}\,\overline{B} + AB.B$

$= (\overline{A} + \overline{B})\overline{B} + AB = \overline{A}\,\overline{B} + AB + \overline{B}$

এই control bit এর জন্য i/p ই A+B দিবো

$(A+B) \oplus 0 = A + B$

$X = A + S_2 \overline{S_1}\,\overline{S_0} B$

2nd case এ:

$(A+K) \oplus \overline{B} = AB$

↳ A এর সাথে এমন কিছু যোগ করবো যেন AB চলে আসে।

$= \overline{(A+K)}\,\overline{B} + (A+K)B = \overline{A}\,\overline{K}\,\overline{B} + AB + KB$

K যদি $\overline{B}$ হয় $\longrightarrow \overline{A}B\overline{B} + AB + B\overline{B} = AB$

	s_2	s_1	s_0	X_i	Y_i	Automatically Obtained F_i	Required F_i
B	1	0	0	A_i	0	$F_i = A_i$	$F_i = A_i + B_i$
$\overline{B}$	1	1	0	A_i	B_i'	$F_i = A_i \odot B_i$	$F_i = A_iB_i$

$$X = A + S_2\overline{S_1}\,\overline{S_0}B + S_2S_1\overline{S_0}\,\overline{B}$$

Incorporating remaining functions

From Table 9-3, we see that when $s_2 = 1$, the input carry C_i in each stage must be 0. With $s_1 s_0 = 00$, each stage as it stands generates the function $F_i = A_i$. To change the output to an OR operation, we must change the input to each full-adder circuit from A_i, to $A_i + B_i$. This can be accomplished by ORing B_i and A_i when $s_2 s_1 s_0 = 100$.

The other selection variables that give an undesirable output occur when $s_2 s_1 s_0 = 110$. The unit as it stands generates an output $F_i = A_i \odot B_i$ but we want to generate the AND operation $F_i = A_i B_i$. Let us investigate the possibility of ORing each input A_i with some Boolean function K_i. The function so obtained is then used for X_i when $s_2 s_1 s_0 = 110$:

$$F_i = X_i \oplus Y_i = (A_i \oplus K_i) \oplus B'_i = A_i B_i + K_i B_i + A'_i K'_i B'_i$$

(A + K)

Careful inspection of the result reveals that if the variable $K_i = B'_i$, we obtain an output:

$$F_i = A_i B_i + B'_i B_i + A_i B_i B'_i = A_i B_i$$

Two terms are equal to 0 because $B_i B'_i = 0$. The result obtained is the AND operation as required. The conclusion is that, if A_i is ORed with B'_i when $s_2 s_1 s_0 = 110$, the output will generate the AND operation.

Final Boolean Functions

- Combining the arithmetic and logical cases, we get the final form of the Boolean function as:

$X_i = A_i + s_2 s_1' s_0' B_i + s_2 s_1 s_0' B_i'$

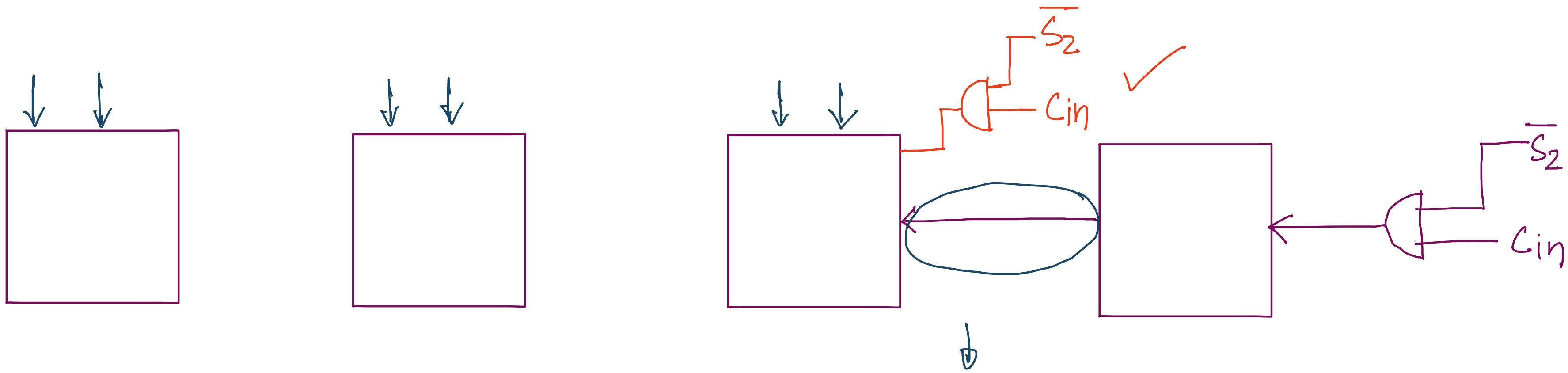
$Y_i = s_0 B_i + s_1 B_i'$

$Z_i = s_2' C_i$

↳ logical operation এর সময় যেন Z_i 0 হয়ে যায়।

প্রতি bit এর জন্য

Selection				Output	Function
s_2	s_1	s_0	C_{in}		
0	0	0	0	$F = A$	Transfer A
0	0	0	1	$F = A + 1$	Increment A
0	0	1	0	$F = A + B$	Addition
0	0	1	1	$F = A + B + 1$	Add with carry
0	1	0	0	$F = A - B - 1$	Subtract with borrow
0	1	0	1	$F = A - B$	Subtraction
0	1	1	0	$F = A - 1$	Decrement A
0	1	1	1	$F = A$	Transfer A
1	0	0	X	$F = A \vee B$	OR
1	0	1	X	$F = A \oplus B$	XOR
1	1	0	X	$F = A \wedge B$	AND
1	1	1	X	$F = \bar{A}$	Complement A



করা যাবে না।

logical এ cin 0 হওয়া লাগবে সবগুলোতেই।

<u>CI তে</u>

X_i
Y_i
Z_i } যেখানে box দেওয়া থাকবে সেখানে final expression লেখা লাগবে।

Let's See Another Example

- Derive the input equations (Xi, Yi and Zi) for the parallel adders to be used in the ALU which satisfies the following functional design specification.

s_2	s_1	c_{in}	Required Functions
0	0	0	F = AB + C
0	0	1	F = AB + C + 1
0	1	0	F = AB
0	1	1	F = AB + 1
1	0	x	F = (AB)'
1	1	x	F = AB

S_2	S_1	C_{in}	X	Y	Z	Required Function
0	0	0	AB	C	0	$AB + C$
0	0	1	AB	C	1	$AB + C + 1$
0	1	0	AB	0	0	AB
0	1	1	AB	0	1	$AB + 1$
1	0	x	AB	C	x	$F = (AB)'$
1	1	x	AB	0	x	$F = AB$

এগুলো বানাতে যা যা লাগবে সেটাই X, Y, Z এ লিখবো।

logical এর জন্য $S_2 = 1$

$S_1 = 0$ তে $Y = C$ } arithmetic এ
$S_1 = 1$ এ $Y = 0$ } পেয়েছি।

arithmetic এ এই দুটো consider করবো না।

$AB \oplus C \longrightarrow (AB)'$

তার মানে 1 পাঠাতে হবে এই control bit এর জন্য।

$S_2 \overline{S_1}$ add করে দিলেই হবে।

<u>arithmetic</u>

$X = AB$

$Y = \overline{S_1} C$

$Z = C_{in} \overline{S_2}$

<u>logical</u>

$X = AB$

$Y = S_2 \overline{S_1}$

$Z = \overline{S_2} C_{in}$

Finally

$X = AB$

$Y = \overline{S_1} C + S_2 \overline{S_1}$

$Z = \overline{S_2} C_{in}$

Solution

s_2	s_1	c_{in}	X	Y	Z	Required Functions
0	0	0	AB	C	0	F= AB + C
0	0	1	AB	C	1	F = AB + C +1
0	1	0	AB	0	0	F = AB
0	1	1	AB	0	1	F = AB + 1
1	0	x	AB	1	x	F = (AB)’
1	1	x	AB	0	x	F = AB

Solution

- $X = AB$
- $Y = s_1' C$
- $Z = s_2' c_{in}$

- Then for logical operations,
  - $X = AB$
  - $Y = s_1' C + s_2 s_1'$
  - $Z = s_2' c_{in}$