DIGITAL ARITHMETIC
Miloš D. Ercegovac and Tomás Lang
Morgan Kaufmann Publishers, an imprint of Elsevier, ©2004
– Updated: February 15, 2005 –

Chapter 2: Solutions to Selected Exercises

– with contributions by Elisardo Antelo –

Exercise 2.1

Assuming that c_i is connected to the XOR input with load factor 1.1 (Fig. 2.5(c)), the average delay of the carry-out is

$$T_1 = t_{NAND}(1) + t_{NAND}(2.1) = 0.07 + 0.033 + 0.07 + 0.033 \times 2.1 = 0.242ns$$

Adding an inverter and changing the XOR into XNOR, we obtain for the carry delay:

$$T_2 = t_{NAND}(1) + t_{NAND}(2) = 0.239ns$$

This represents a 1.4% reduction in the carry delay. Note that the difference is very small because of the XOR input with load factor 1.1. A larger reduction would result if the XOR input load factors were symmetrical at 2.

Exercise 2.4

$T_{SRA} = t_{sw} + (n-1)t_p + (n/m)t_{buf} + t_s$ (Expression (2.27))

$t_{sw} = max(t_{gi}, t_{ki}, t_{pi}) + t_{NAND-2(L=2)} = t_{pi} + t_{NAND-2(L=2)} = 0.329 + 0.136 = 0.465ns$

where, assuming a switch has one standard load,

$t_{gi} = t_{AND-2} = 0.16 + 0.027 \times 1 = 0.187ns$

$t_{ki} = t_{NOR-2} = 0.07 + 0.046 \times 1 = 0.116ns$

$t_{pi} = t_{XOR-2} = 0.30 + 0.029 \times 1 = 0.329ns$

$t_p = t_{NAND-2} = 0.07 + 0.033 \times 2 = 0.136ns$ $(L = 2)$

$t_{buf} = 1.5 \times 0.136 = 0.204$

$t_s = 0.46 + 0.03 \times L = 0.46ns$ (Table 2.2, delay c_i to s_i with $L = 0$)

Therefore,

$T_{SRA} = 0.465 + 31 \times 0.136 + 8 \times 0.204 + 0.46 \approx 6.8ns$

From Exercise 2.2, $T_{CRA} = 13.8ns$ so the SRA aproximately halves the delay. Note that to reduce the load the network for computing the sum bits uses separately obtained p_i signals

Exercise 2.5

Figure E2.5 shows the carry chains for the given operands.

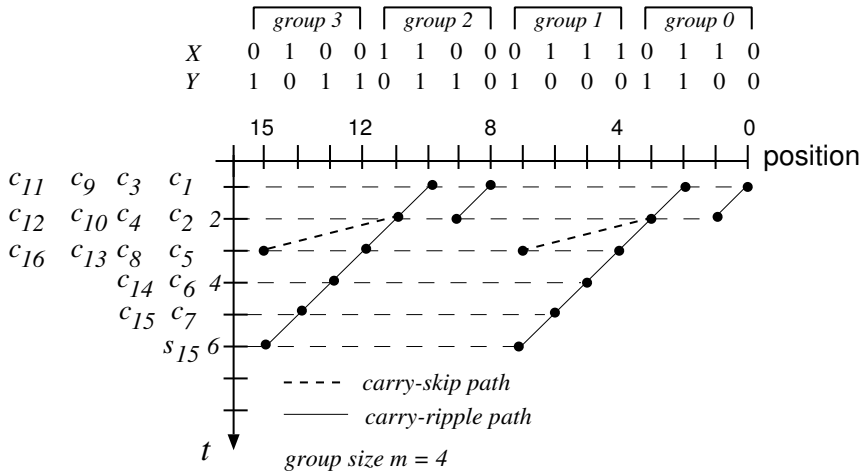

Figure E2.5: Carry chains in carry-skip adder (Exercise 2.5).

Note that c_8 is obtained via the skip path and via the carry-ripple path in the groups 0 and 1 after a delay of 3 units.

Exercise 2.10

(a) $T = mt_c + (s-1)t_{mux} + (p-2)t_{mux} + (s-1)t_{mux} + (m-1)t_c + t_s$.

(b) Let $t_c = t_{mux}$ and $m = s$. $T = (4m - 3 + n/m^2)t_c + t_s$ and $m_{opt} = (n/2)^{1/3}$.

Exercise 2.13

The g_i and a_i signals are

x	0	1	0	1
y	1	0	0	1
g_i	0	0	0	1
a_i	1	1	0	1

The expressions and values for the CLG-4 carries are

$$
\begin{aligned}
c_0 &= 1 \\
c_1 &= g_0 + a_0c_0 = 1 + 1 \cdot 1 = 1 \\
c_2 &= g_1 + a_1g_0 + a_1a_0c_0 = 0 + 0 \cdot 1 + 0 \cdot 1 \cdot 1 = 0 \\
c_3 &= g_2 + a_2g_1 + a_2a_1g_0 + a_2a_1a_0c_0 = 0 + 1 \cdot 0 + 1 \cdot 0 \cdot 1 + 1 \cdot 0 \cdot 1 \cdot 1 \cdot 1 = 0 \\
c_4 &= g_3 + a_3g_2 + a_3a_2g_1 + a_3a_2a_1g_0 + a_3a_2a_1a_0c_0 \\
&= 0 + 1 \cdot 0 + 1 \cdot 1 \cdot 0 + 1 \cdot 1 \cdot 0 \cdot 1 + 1 \cdot 1 \cdot 0 \cdot 1 \cdot 1 = 0
\end{aligned}
$$

Exercise 2.15

A 64-bit, three-level carry-lookahead adder is shown in Figure E2.15.

Exercise 2.17

$n = 128$, $m = 4$, $t_{clg} = t_{AG} = 6t_{ag} = 3t_s$

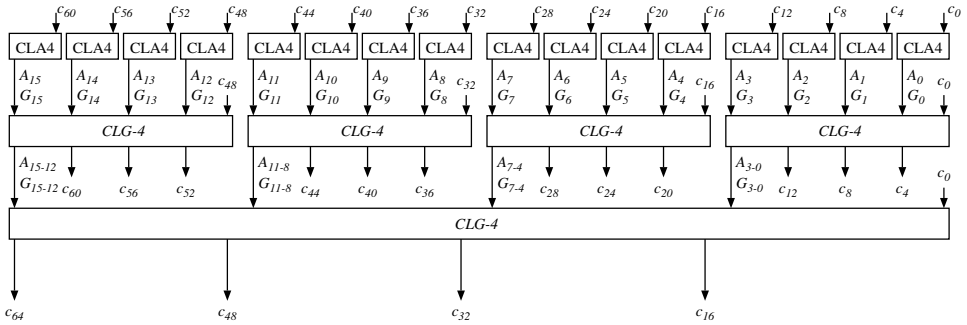

Figure E2.15: 64-bit three-level carry-lookahead adder.

$$T_{1-CLA} = t_{ag} + (n/m)t_{clg} + t_s = 1 + 32 \times 6 + 2 = 195t_{ag}$$

$$T_{2-CLA} = t_{ag} + t_{AG} + (n/m^2)t_{clg} + t_{clg} + t_s = 1 + 6 + 8 \times 6 + 6 + 2 = 63t_{ag}$$

$$T_{3-CLA} = t_{ag} + 2t_{AG} + (n/m^3)t_{clg} + 2t_{clg} + t_s = 1 + 12 + 12 + 12 + 2 = 39t_{ag}$$

For the 4-level CLA we use another level with a group size of 2. Because of the smaller size of this group the delay of this level is smaller, we assume it to be $t_{clg2} = 2t_{a,g}$.

$$T_{4-CLA} = t_{ag} + 3t_{AG} + t_{clg2} + 3t_{clg} + t_s = 1 + 18 + 2 + 18 + 2 = 41t_{ag}$$

Exercise 2.20

i	8	7	6	5	4	3	2	1	0
x_i		0	1	0	1	0	1	1	1
y_i		1	1	1	0	0	1	1	1
g_i		0	1	0	0	0	1	1	1
a_i		1	1	1	1	0	1	1	1
p_i		1	0	1	1	0	0	0	0

Level 1 outputs:

$$\begin{aligned}
g_{(0,-1)} &= 1 = c_1 \\
g_{(1,0)} &= g_1 + a_1 g_0 = 1, \quad a_{(1,0)} = a_1 a_0 = 1 \\
g_{(2,1)} &= g_2 + a_2 g_1 = 1, \quad a_{(2,1)} = a_2 a_1 = 1 \\
g_{(3,2)} &= g_3 + a_3 g_2 = 0, \quad a_{(3,2)} = a_3 a_2 = 0 \\
g_{(4,3)} &= g_4 + a_4 g_3 = 0, \quad a_{(4,3)} = a_4 a_3 = 0 \\
g_{(5,4)} &= g_5 + a_5 g_4 = 0, \quad a_{(5,4)} = a_5 a_4 = 1 \\
g_{(6,5)} &= g_6 + a_6 g_5 = 1, \quad a_{(6,5)} = a_6 a_5 = 1 \\
g_{(7,6)} &= g_7 + a_7 g_6 = 1, \quad a_{(7,6)} = a_7 a_6 = 1
\end{aligned}$$

Level 2 outputs:

$$
\begin{array}{lcl}
g_{(1,-1)} & = & g_{(1,0)} + a_{(1,0)}c_0 = 1 = c_2 \\
g_{(2,-1)} & = & g_{(2,1)} + a_{(2,1)}g_{(0,-1)} = 1 = c_3 \\
g_{(3,0)} & = & g_{(3,2)} + a_{(3,2)}g_{(1,0)} = 0, \quad a_{(3,0)} = a_{(3,2)}a_{(1,0)} = 0 \\
g_{(4,1)} & = & g_{(4,3)} + a_{(4,3)}g_{(2,1)} = 0, \quad a_{(4,1)} = a_{(4,3)}a_{(2,1)} = 0 \\
g_{(5,2)} & = & g_{(5,4)} + a_{(5,4)}g_{(3,2)} = 0, \quad a_{(5,2)} = a_{(5,4)}a_{(3,1)} = 0 \\
g_{(6,3)} & = & g_{(6,5)} + a_{(6,5)}g_{(4,3)} = 1, \quad a_{(6,3)} = a_{(6,5)}a_{(4,3)} = 0 \\
g_{(7,4)} & = & g_{(7,6)} + a_{(7,6)}g_{(5,4)} = 1, \quad a_{(7,4)} = a_{(7,6)}a_{(5,4)} = 1
\end{array}
$$

Level 3 outputs:

$$
\begin{array}{lcl}
c_4 & = & g_{(3,0)} + a_{(3,0)}c_0 = 0 \\
c_5 & = & g_{(4,1)} + a_{(4,1)}g_{(0,-1)} = 0 \\
c_6 & = & g_{(5,2)} + a_{(5,2)}g_{(1,-1)} = 0 \\
c_7 & = & g_{(6,3)} + a_{(6,3)}g_{(2,0)} = 1
\end{array}
$$

Level 4 outputs:

$$
\begin{array}{lcl}
s_0 & = & p_0 \oplus c_0 = 1 \\
s_1 & = & p_1 \oplus c_1 = 1 \\
s_2 & = & p_2 \oplus c_2 = 1 \\
s_3 & = & p_3 \oplus c_3 = 1 \\
s_4 & = & p_4 \oplus c_4 = 1 \\
s_5 & = & p_5 \oplus c_5 = 1 \\
s_6 & = & p_6 \oplus c_6 = 0 \\
s_7 & = & p_7 \oplus c_7 = 0 \\
c_8 & = & g_{(7,0)} + a_{(7,0)}c_0 = 1
\end{array}
$$

Exercise 2.23

A diagram of a 4-bit conditional-adder module is shown in Figure E2.23.

Exercise 2.26

X	01	01	01	11
Y	10	10	11	11
S^0	11	11	00	10
c^0	0	0	1	1
S^1	00	00	01	11
c^1	1	1	1	1
S^0	11	11	01	10
c^0	0		1	
S^1	00	00	01	11
c^1	1		1	
S^0	00	00	01	10
c^0	1			
S^1	00	00	01	11
c^1	1			

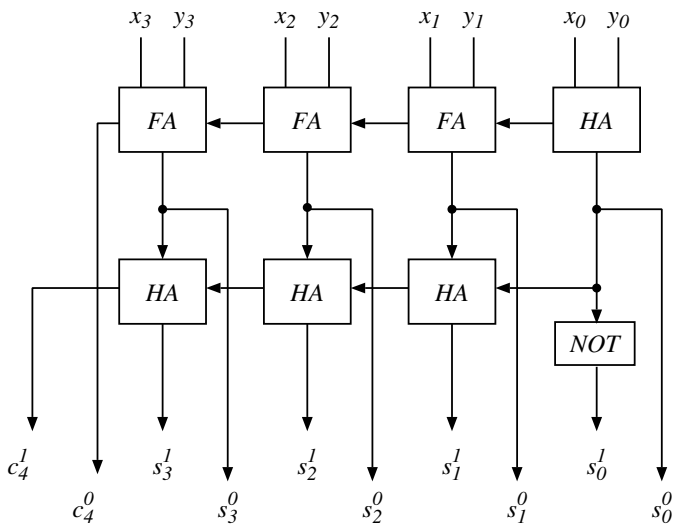

Figure E2.23: 4-bit conditional adder for Exercise 2.23.

The result is (c^0, S^0) because $c_{in} = 0$.

Exercise 2.29

a) Type 1 adder:

x	1000	100	111
y	0111	000	110
c_i^0	11111	110	011
c_i^1	00000	001	100
c_i	00000	001	100
s_i	01111	101	101

The actual delay, assuming critical path in producing F, is

$$T_{Type1} = t_{XOR} + t_{OR-2} + 10 \times t_c + t_{OR-2}$$

where

t_{XOR} is the time to get p_i (g_i and k_i are obtained with less delay);

t_{OR-2} is the time to get $(p_i + k_i)$ and $(p_i + g_i)$;

t_c is the delay of producing a carry:

$$t_c = t_{AND-2} + t_{OR-2}$$

t_{OR-2} is the time to get the completion signal F.

Given that t_c has the same expression for the carry-ripple adder and that the actual delay of t_c is 15% smaller than its worst-case delay and assuming the same variation for t_{XOR} and t_{OR-2}, we get:

$$T_{Type1} \approx 0.85 T_{CRA}$$

b) Type 2 adder:

x	1000100111
y	0111000110
chains	jihgfedcba
timing	6543211111

In this example, the longest chain is zero-carry chain efghij of 6 positions.

The actual delay is

$$T_{Type2} = t_{XOR} + t_{max} + t_{OR-2} + t_{AND-10}$$

where $t_{max} = 6t_c$.

Consequently, including the delay of AND-10, for this input pattern the addition delay is rougly 70% of that of the adder of type I.

Exrecise 2.32

a)

X		0	1	1	1	1	1	0	0	0	0	1	1	0	0	1	1
Y		1	1	1	0	0	0	1	0	0	1	1	0	1	1	0	1
W		1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
$S*$		0	0	1	1	0	1	0	0	1	1	1	1	0	1	0	0
$C*$	1	1	1	0	1	0	1	0	0	0	1	0	1	0	1	1	0a
Z		1	0	1	0	1	1	1	1	0	1	1	1	0	1	1	1
S	1	0	1	0	0	1	1	1	1	1	1	0	1	0	1	0	1
C	1	0	1	1	0	1	0	0	0	1	1	1	0	1	1	0	

a carry in

b)

X		0	1	1	1	1	1	0	0	0	0	1	1	0	0	1	1
Y		1	1	1	0	0	0	1	0	0	1	1	0	1	1	0	1
W		1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
Z		1	0	1	0	1	1	1	1	0	1	1	1	0	1	1	1
T	1	1	1	0	1	0	1	0	0	0	1	0	1	0	1	1	0a
P		1	0	0	1	1	0	1	1	1	0	0	0	0	0	1	1
S	1	0	1	0	0	1	1	1	1	1	1	0	1	0	1	0	1
C	1	0	1	1	0	1	0	0	0	1	1	1	0	1	1	0	

a carry in; P output of Odd-parity module.

Exercise 2.35

	101	110	110	011
	1	1	0	1
	011	100	111	011
	001	011	101	111
1	1	1	0	0

Exercise 2.39

Method 1:

X			0	1	$\bar{1}$	1	$\bar{1}$	0	1	$\bar{1}$
Y			1	0	1	0	1	$\bar{1}$	$\bar{1}$	1
H		1	1	0	1	0	0	0	0	
Z			$\bar{1}$	$\bar{1}$	0	$\bar{1}$	0	$\bar{1}$	0	0
Q		1	0	$\bar{1}$	1	$\bar{1}$	0	$\bar{1}$	0	0
T	0	0	$\bar{1}$	0	$\bar{1}$	0	$\bar{1}$	0	0	
W		1	0	1	1	1	0	1	0	0
S		1	$\bar{1}$	1	0	1	$\bar{1}$	1	0	0

Method 2:

X			0	1	$\bar{1}$	1	$\bar{1}$	0	1	$\bar{1}$
Y			1	0	1	0	1	$\bar{1}$	$\bar{1}$	1
P			0	0	1	0	1	1	1	1
T		1	0	0	0	0	$\bar{1}$	0	0	
W			$\bar{1}$	1	0	1	0	1	0	0
S		1	$\bar{1}$	1	0	1	$\bar{1}$	1	0	0

Exercise 2.43

Radix-2 signed digit addition of one conventional and one signed-digit operand:

X			0	1	1	1	0	1	1	0
Y^+			1	0	1	0	0	0	1	1
Y^-			0	1	0	0	0	1	0	0
W			1	0	0	1	0	0	0	1
T		1	0	1	1	0	0	1	1	
S^+		1	0	1	1	0	0	1	1	0
S^-			1	0	0	1	0	0	0	1
S		1	$\bar{1}$	1	1	$\bar{1}$	0	1	1	$\bar{1}$