

Topics

- Arithmetic Circuits

17 March 2009

1

Building Blocks for Digital Architectures

- **Arithmetic unit**
 - Bit-sliced datapath (adder, multiplier, shifter, comparator, etc.)
- **Memory**
 - RAM, ROM, Buffers, Shift registers
- **Control**
 - Finite state machine
 - Counters
- **Interconnect**
 - Switches, Arbiters, Bus

17 March 2009

2

Datapath operators

- Operate on multiple bits at once
- Allows tight design of several elements
 - Arithmetic/logic functions, shifters
 - Registers, Register Files
 - Wires, buses

17 March 2009

3

Control

- Control signals determine the flow of data along the datapath
- Functional control of data
 - Whether to add or subtract
 - When to store to what register
 - Which wires to transmit data on

17 March 2009

4

Datapath and Control

- Separation of datapath and control simplifies design
- Layout strategy
 - Datapath horizontal in metal2
 - Control vertical in metal3

17 March 2009

5

Datapath operators

- **Half Adder**

“half” means no carry-in

Input: A, B

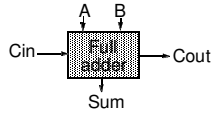
Sum: $s = A \oplus B$

Carry out: $c = AB$

17 March 2009

6

Datapath operators



A	B	C _i	S	C _o
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

17 March 2009

7

Datapath operators

- Full Adder

$$Sum = A \oplus B \oplus C$$

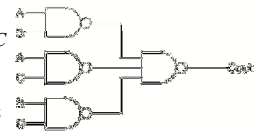


$$Cout = AB + BC + AC$$

$$= \overline{AB} \cdot \overline{BC} \cdot \overline{AC}$$

$$\overline{Cout} = \overline{AB + BC + AC}$$

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



17 March 2009

8

Datapath operators

- Full Adder

$$Sum = A \oplus B \oplus C$$

$$= ABC + ABC + ABC + ABC$$

$$= ABC + (A + B + C)(AB + BC + AC)$$

$$= ABC + (A + B + C)(A + B)(B + C)(A + C)$$

$$= ABC + (A + B + C)(AB + BC + AC)$$

$$= ABC + (A + B + C)(Cout)$$

17 March 2009

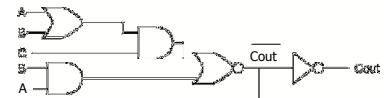
9

Datapath operators

- Full Adder

$$\overline{Cout} = \overline{AB + BC + AC}$$

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



$$Sum = ABC + (A + B + C)(\overline{Cout})$$



17 March 2009

10

Datapath operators

- Full Adder

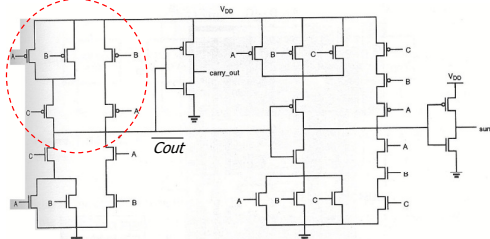


Fig. 7.26, "CMOS Digital Integrated Circuits", Kang and Leblebici

$$\overline{Cout} = \overline{AB + BC + AC}$$

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

$$ABC + (A + B + C)(\overline{Cout})$$

17 March 2009

11

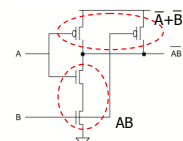
Datapath operators

$$\begin{aligned} \overline{AB + BC + AC} &= \overline{AB} \cdot \overline{BC} \cdot \overline{AC} \\ &= \overline{(A + B)(B + C)(A + C)} \\ &= \overline{AB + BC + AC} \end{aligned}$$

- Equivalent duals

- Allows much simpler layout because the pull-down and pull-up networks are the same

- Take a look at NAND:

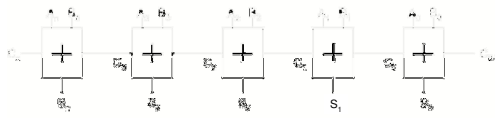


17 March 2009

12

Datapath operators

- Ripple carry adder



- Carry delay increases with number of bits

17 March 2009

15

Datapath operators

- Carry look-ahead adders
 - Calculate carry in parallel with sum
 - Each carry bit is dependent on inputs only - not previous carry

$$S_2 = A_2 \oplus B_2 \oplus C_1$$

$$C_2 = A_2 B_2 + (A_2 \oplus B_2) C_1$$

$$S_3 = A_3 \oplus B_3 \oplus C_2$$

$$C_3 = A_3 B_3 + (A_3 \oplus B_3) C_2$$

17 March 2009

14

Datapath operators

- Carry look-ahead adders

$$G_i = A_i B_i \quad \text{Carry Generate Logic}$$

$$P_i = (A_i \oplus B_i) \quad \text{Carry Propagate Logic}$$

$$C_{i+1} = G_i + P_i C_i$$

$$C_1 = G_0 + P_0 C_0$$

$$C_2 = G_1 + P_1 C_1 = G_1 + P_1 G_0 + P_1 P_0 C_0$$

$$C_3 = G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_0$$

$$C_4 = G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0 + P_3 P_2 P_1 P_0 C_0$$

17 March 2009

15

Datapath operators

- Carry look-ahead adders
 - For more than 4 bits, the effect of fanout can be significant, the delay through the Carry Generate Logic could be more than the carry ripple delay
 - Usually divide the data into four bit chunks with a hierarchy of Propagate and Generate circuitry

17 March 2009

16

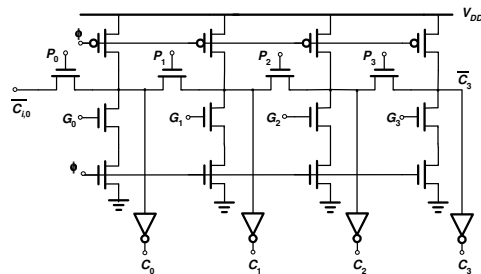
Manchester Carry Chain

- Precharged Carry Chain which uses P and G signals
- Propagate signal connects adjacent carry bits
- Generate signal discharges carry bit
- Worst-case discharge path goes through entire carry chain.

17 March 2009

17

Manchester Carry Chain

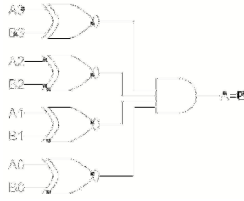


17 March 2009

18

Datapath operators

- Comparators



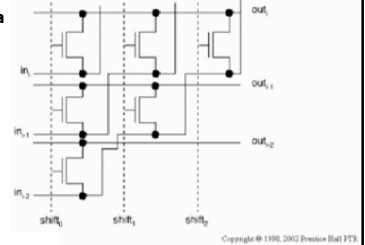
17 March 2009

19

Datapath operators

Barrel Shifters:

- Perform n-bit shift in a single cycle
- Two dimensional array
- Input data travels diagonally upward
- Output wires travel horizontally
- Control signals run vertically

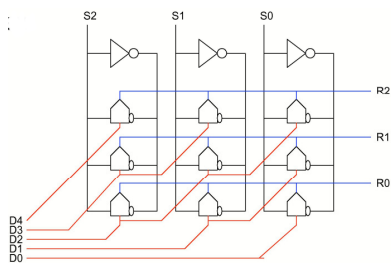


17 March 2009

20

Datapath operators

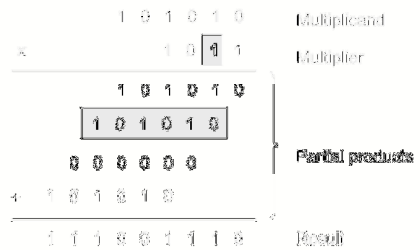
- Shifters



17 March 2009

21

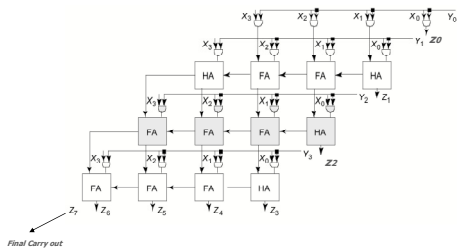
Binary Multiplication



17 March 2009

22

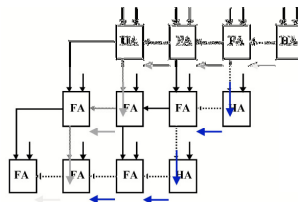
The Array Multiplier



17 March 2009

23

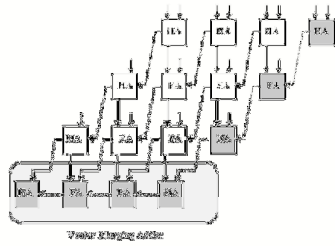
The MxN Array Multiplier — Critical Path



17 March 2009

24

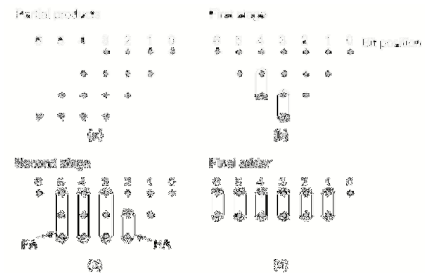
Carry-Save Multiplier



17 March 2009

25

Wallace-Tree Multiplier



17 March 2009

26