

**University of California at Berkeley**  
**College of Engineering**  
**Department of Electrical Engineering and Computer Sciences**

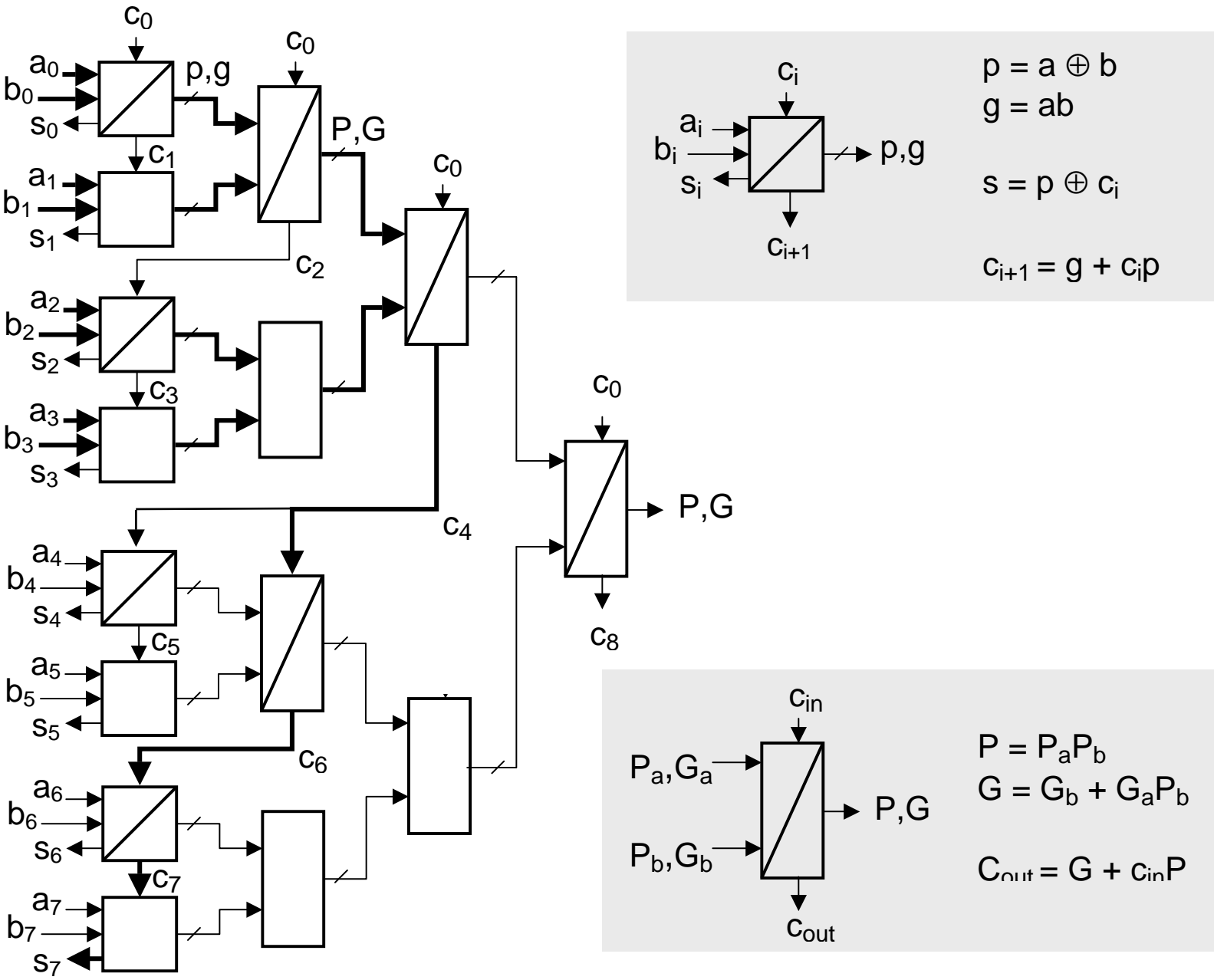
EECS150  
Spring 2002

J. Wawrzynek

Homework #5

This homework is due on **Friday March 8<sup>th</sup> by 1pm**. Homework will be accepted in the EECS150 box on the door to room 218 Cory Hall. Late homework will be penalized by 50%. No late homework will be accepted after the solution is posted.

- Below is the illustration of the 8-bit carry look-ahead adder presented in class. Draw a line on the circuit representing the critical path (signal path with worst-case delay). Using “big O” notation, how does the delay and cost scale with n for this type of adder? **Delay is  $O(\log_2 n)$**



2. Carry select adders.

- a) In class we discussed the carry-select adders. Assuming that the select groups are all of the same number of bits, what is the optimal size select group for a 32-bit adder?

**Assuming the delay of a mux is the same as a full-adder cell.**

$$\text{Delay} = \text{block\_size} + \lceil 32/\text{block\_size} \rceil - 1$$

**Min Delay of 11 units occurs when block\_size = (4 to 8)**

- b) In class we also discussed the possibility of applying the carry-select idea hierarchically. The idea is that a ripple adder of n-bits can be split and implemented as a carry-select adder with group size n/2 (implemented as three ripple adders of size n/2 along with muxes). Then each of the three adders of size n/2 could be split and implemented as a carry-select adder with group size n/4, etc. By continuing the process eventually the adder size would be 1-bit and the process ends.

Discuss the worst case delay through this adder. Again using “big O” notation, how does the delay scale with n? How does the cost (amount of hardware - number of gates or transistors) scale with n?

**The total delay at each stage is the delay of the carry out of the sub-block plus the delay of the mux to the carry out. So the total delay of the circuit is the mux\_delay x (number of stages) which is  $O(\log_2)$ .**

$$\text{Area}(n\_bit) = 3 \times \text{Area}(n\_bit/2) = 3^{\log_2(n)}$$

3. For this problem assume that your complete library of logic components comprises 2-input AND, OR, and XOR gates and inverters. Assume that inverters have cost of one unit and delay of one unit, and all other logic gates have cost of 2 units and delay of 2 units (ignore fanout and wire delay). With these assumptions, a 2-input mux would be implemented with two AND gates, one OR gate, and one inverter, and would cost 7 and have delay 5. Generate a table comparing four different adder architectures for delay and cost. The adder architectures are i) 32-bit ripple adder, ii) 32-bit carry-select adder from problem 2.a) above, iii) 32-bit hierarchical carry-select adder from problem 2.b) above, and iv) a 32-bit carry look-ahead adder based on the 8-bit carry look-ahead adder presented in class. Can you draw any conclusions from the results presented in the table?

32-bit adders	Delay	Cost
Ripple	128	320
Carry select	53	623 (block size 8)
Hierarchical carry select	39	3539
Carry look ahead	36	621

4. From Mano, exercises

4-9 input: (ABCD) output: (abcdefg)

$$a = A'C + A'BD + B'C'D' + AB'C$$

$$b = A'B' + A'C'D' + A'CD + AB'C'$$

$$c = A'B + A'D + B'C'D' + AB'C' \quad d = A'CD' + A'B'C + B'C'D' + AB'C' + A'BC'D$$

$$e = A'CD' + B'C'D'$$

$$f = A'BC' + A'C'D' + A'BD' + AB'C'$$

$$g = A'CD' + A'B'C + A'BC' + AB'C'$$

there are 6 common terms:  $B'C'D'$ ,  $AB'C'$ ,  $A'C'D'$ ,  $A'CD'$ ,  $A'B'C$ ,  $A'BC'$

4-10 input: (ABCD) output: (wxyz)

$$w = A'(B+C+D) + AB'C'D' = A \oplus (B+C+D)$$

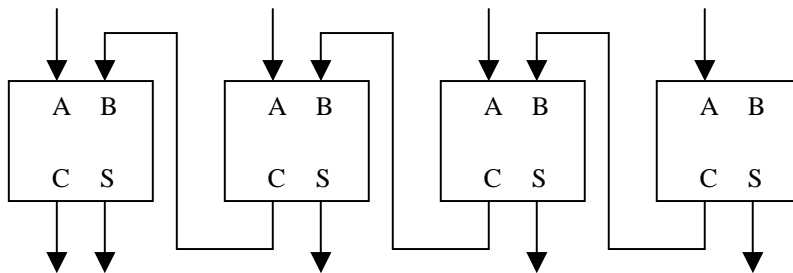
$$x = B'(C+D) + BC'D = B \oplus (C+D)$$

$$y = CD' + C'D = C \oplus D$$

$$z = D$$

For 5-bit complementer with input E and output v.  $v = E \oplus (A+B+C+D)$

4-11



4-13

	SUM	C	V
(a)	1101	0	1
(b)	0001	1	1
(c)	0101	1	0
(d)	1011	0	1
(e)	1111	0	0

4-14

XOR AND OR XOR

$$20 + 10 + 10 + 20 = 60\text{ns}$$

4-15

$$C_4 = G_3 + P_3 C_3$$

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

4-21.

$$(A_0 \oplus B_0)' (A_1 \oplus B_1)' (A_2 \oplus B_2)' (A_3 \oplus B_3)'$$