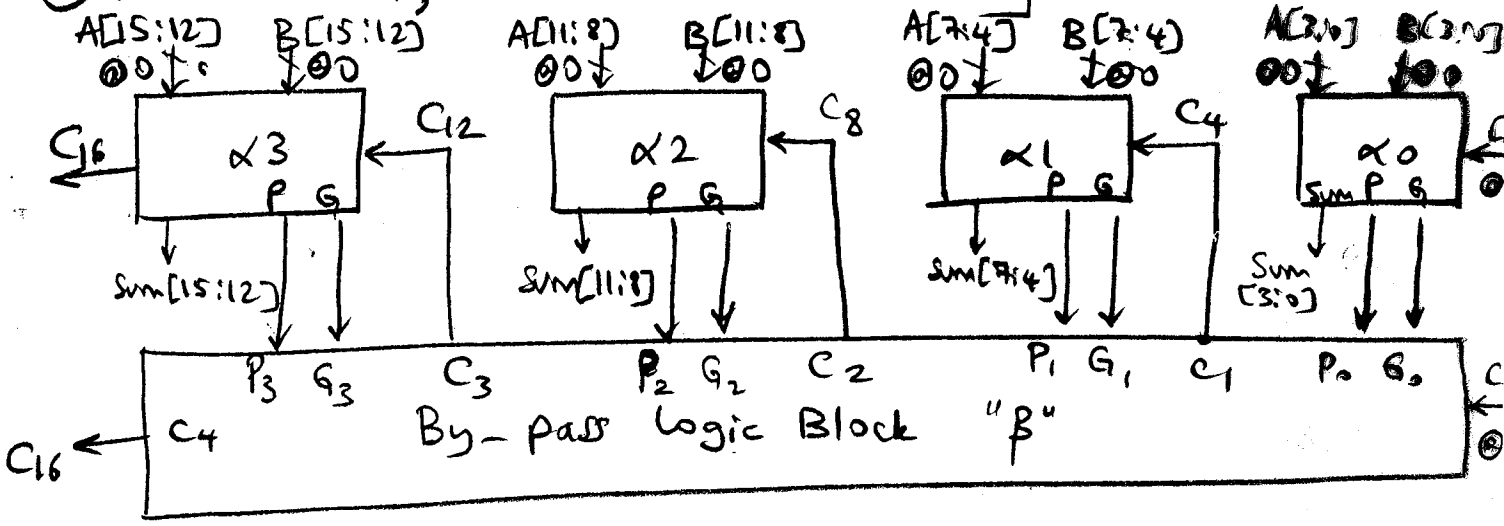
LECTURE: "ADDERS" #2:

AIM: Analyze the delay of a new type of adder. (Variations on what we have seen so far)

① [Problem #4, Fall 2004 Final Exam]



- By-pass logic Block Equations:

$$C_{i+1} = G_i + P_i \cdot C_i \quad , \quad 0 \le i \le 3$$

(Like the carry's are not substituted ∴ not lookahead)

- Each α block is a 4-bit ripple carry adder (RCA)

Q: What is the worst-case delay of the above 16-bit adder?

Remarks: The bypass logic contains less hardware than the carry lookahead logic. Each carry depends on the previous ("macro") carry only. ∴ Expect the performance (delay) to be worse than that of the CLA block.

DELAY ANALYSIS:

① The group propagate of each 4-bit adder is @2 (since $P_{group} = P_0 P_1 P_2 P_3$, @1 @1 @1 @1, just like before) and the group generate of each 4-bit adder is @3

② Hence, inside the bypass logic block.

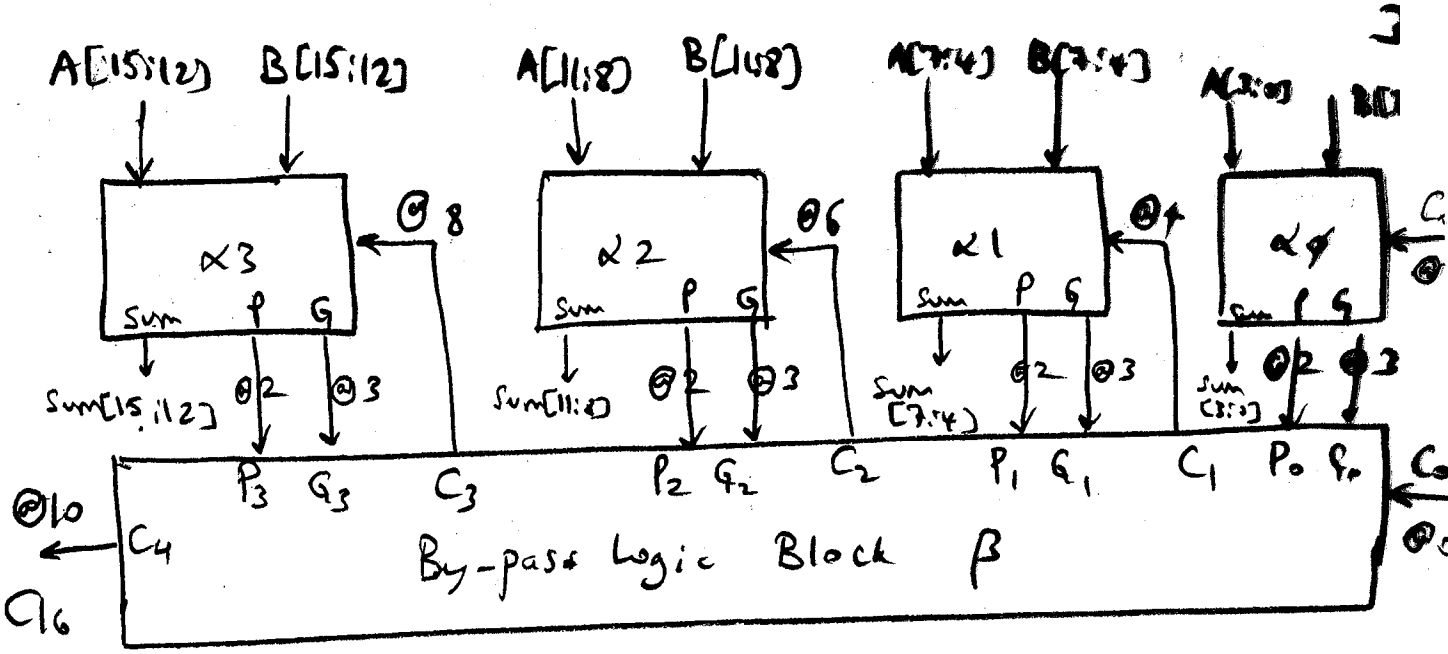
$$C_1 = G_0 + P_0 \cdot C_0 \quad (@4 = @3 + @2 \cdot @0)$$

$$C_2 = G_1 + P_1 C_1 \quad (@6 = @3 + @2 \cdot @4,\ @5)$$

Note this!

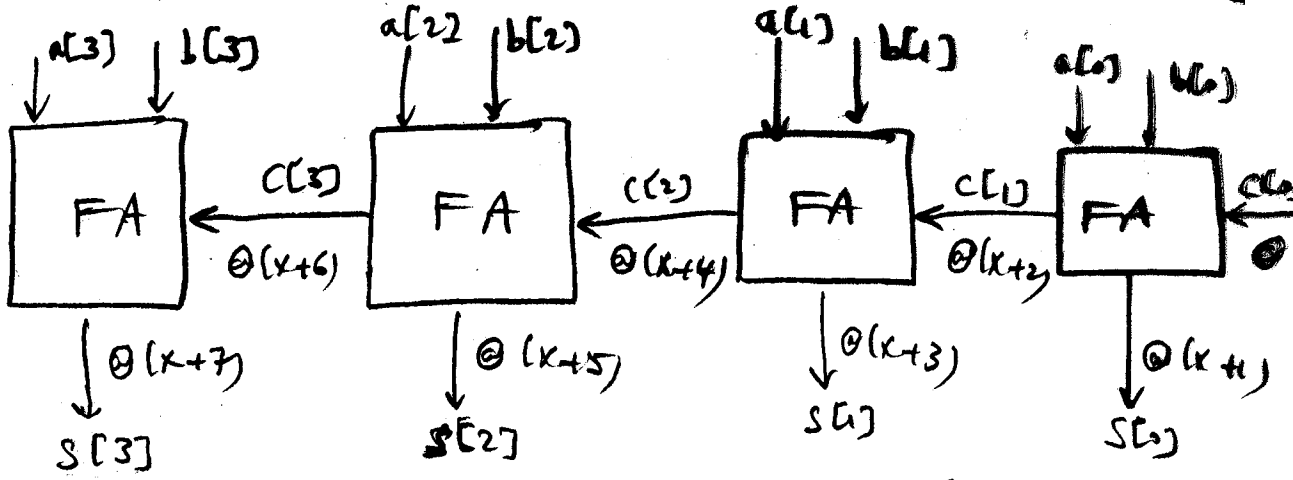
$$C_3 = G_2 + P_2 C_2 \quad (@8 = @3 + @2 \cdot @6,\ @7)$$

$$C_4 = G_3 + P_3 C_3 \quad (@10 = @3 + @2 \cdot @8,\ @9)$$



- The worst-case delay is the worst-case delay of the Sum vector (if this is a stand-alone 16-bit adder)
- The SUM[15:12] will become available last.
- ∴ need to compute the delay of the 4-bit ripple carry adder

③

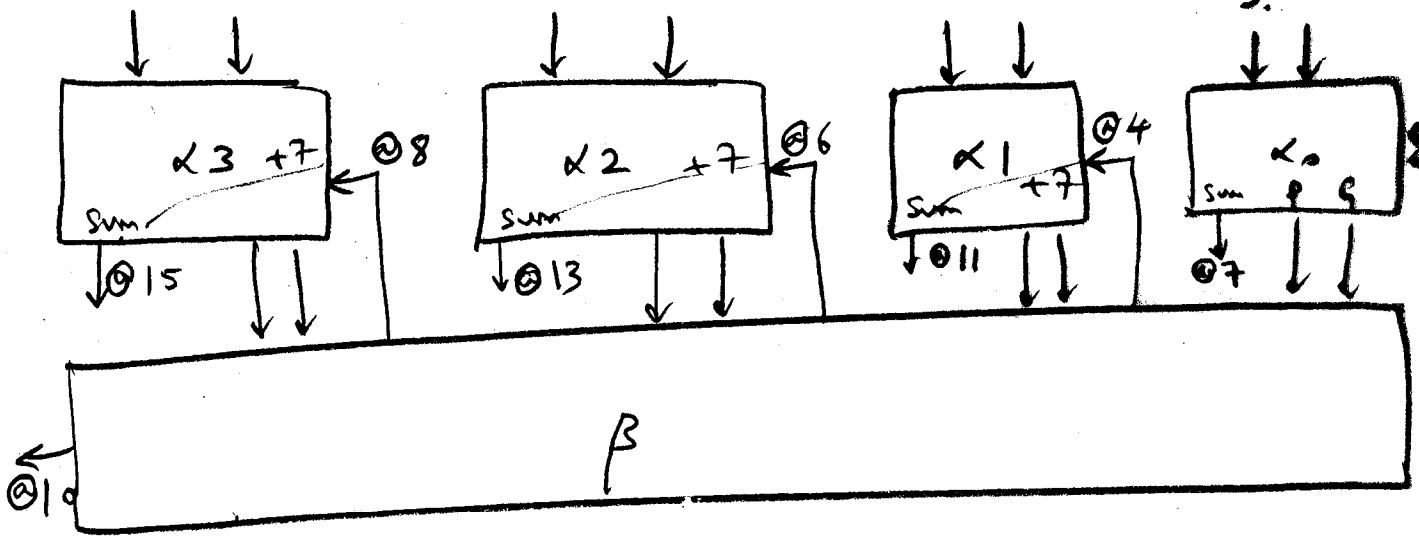


$$C[i+1] = \underbrace{a[i]}_{@0} \cdot \underbrace{b[i]}_{@0} + \underbrace{(\underbrace{a[i]}_{@0} \oplus \underbrace{b[i]}_{@0})}_{@1} \cdot \underbrace{c[i]}_{@x_i} \qquad @(x_i+2)$$

$$S[i] = \underbrace{a[i] \oplus b[i]}_{@1} \oplus \underbrace{c[i]}_{@x_i} \qquad @(x_i+1)$$

∴ The sum of the 4-bit ripple carry adder will be available @(x+7) if its C[0] is available @ x.

Hence, the final answer is: 15 : worst-case delay.

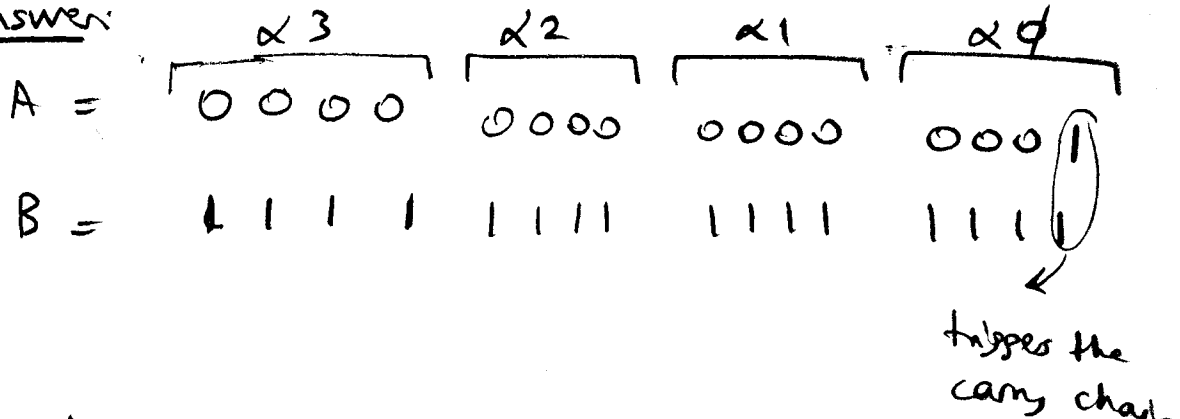


(b) What is the critical path of this adder?

Answer: The path through the bypass logic carry chain of the first three stages plus the ripple carry chain of the last stage ($\alpha 3$).

(c) A pair that achieves this worst-case delay?

Answer:



- Assures that $\alpha 1$ and $\alpha 2$ do not group generate.
- The fact that the $\alpha 0$ stage group generates is OK since it is the first-stage and its carryout is still available @ 4

(d) Explain how 'grop generate' functionality affects the worst-case delay of this adder.

Answer: It does not.

- grop generate lowers (average delay / best-case delay)
- worst-case delay: critical inputs do not use grop generate

Problem 4 - Part B:

- Instead of the 4-bit RCA, assume we use 4-bit CLA adders.
- How does the critical path change?

Answer: Same critical path through first 3 stages of lookahead.

But: last stage will be delay of CLA, not RCA

$$\therefore \underbrace{8}_{(C_{12})} + \underbrace{3}_{\alpha 3\ \text{CLA}} = \boxed{11} \quad \text{new worst-case delay}$$

($\alpha 3$)

∴ only difference in speed is that we get a speed-up in $\alpha 3$ to compute $sum[5:12)$ from C_{12}.