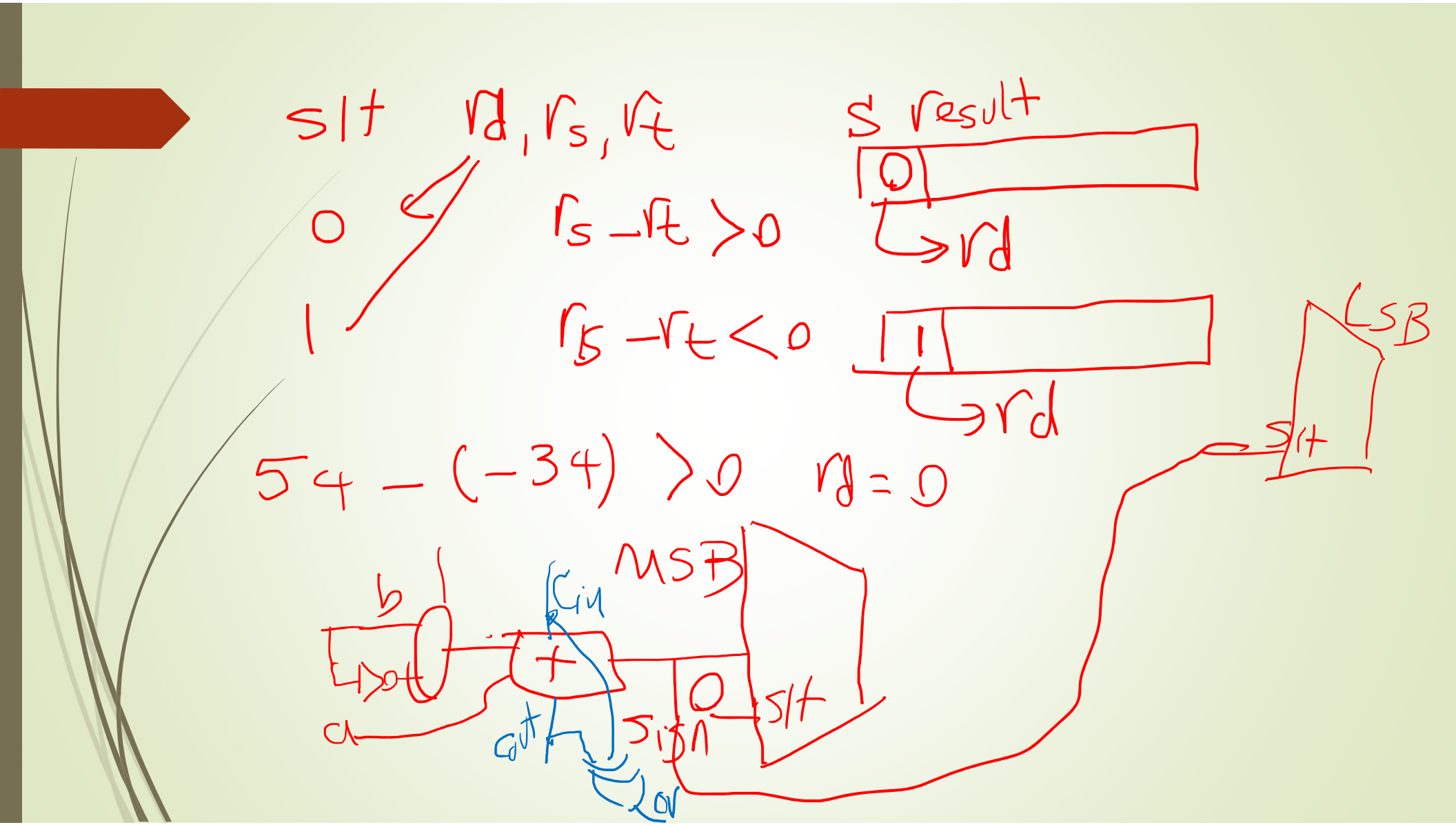
EGC442
Class Notes
3/3/2023

Baback Izadi
Division of Engineering Programs
bai@engr.newpaltz.edu

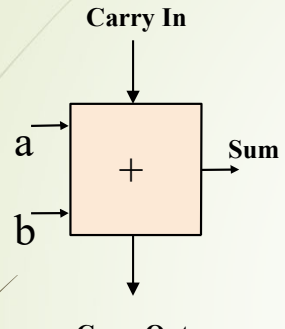
Test 1:

- Chapter 2
  - Performance problems
- Chapter 3
  - MIPS instruction set
  - C to MIPS
  - MIPS to C
  - MIPS to machine code
- Chapter 4
  - Hardware and algorithm for Multiplication
  - Floating Point
  - ALU design


slt rd, rs, rt
0
1
rs - rt > 0
rs - rt < 0
S result
0
rd
1
rd
LSB
slt
54 - (-34) > 0
rd = 0
b
MSB
Cin
+
0
slt
sign
Cout
Cout

Making a faster adder Full Adder

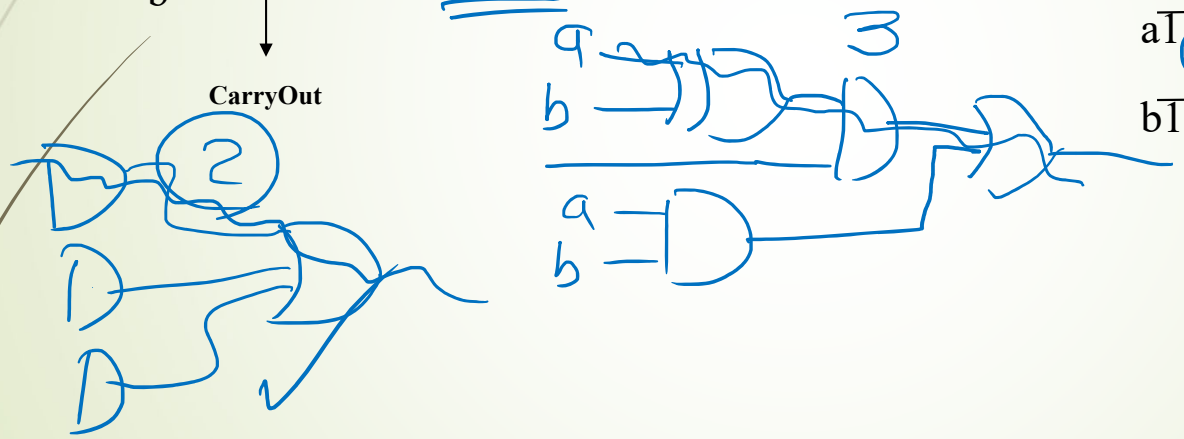
Let's look at a 1-bit ALU for addition:

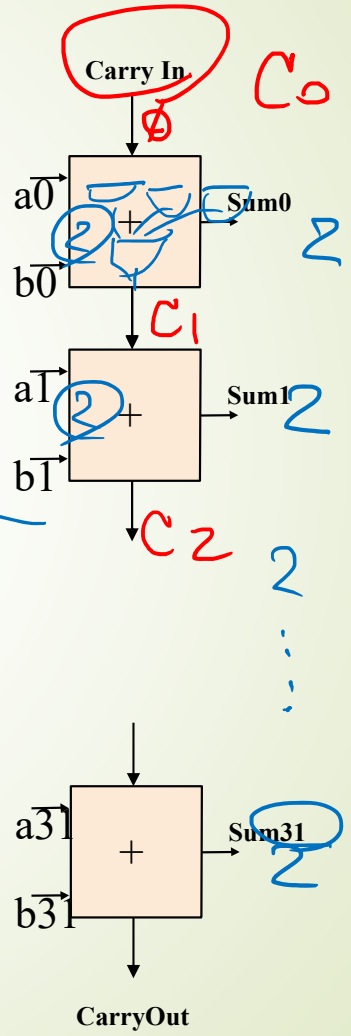


$$Sum = a \oplus b \oplus c_{in}$$

~~$C_{out} = a\,b + (a \oplus b)\,c_{in}$~~

$$C_{out} = a\,b + a\,c_{in} + b\,c_{in}$$





What is the propagation delay of a 32-bit adder?

32 * 2 = 64

Problem with Ripple Carry

- Is a 32-bit ALU as fast as a 1-bit ALU?
- Is there more than one way to do addition?
  - two extremes: ripple carry and sum-of-products
- Can you see the ripple? How could you get rid of it?

$c_1 = b_0c_0 + a_0c_0 + a_0b_0$

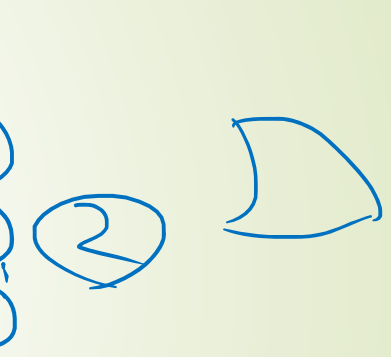
$c_2 = b_1c_1 + a_1c_1 + a_1b_1$ $\quad c_2 =$

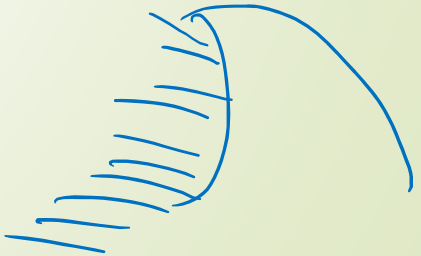
$c_3 = b_2c_2 + a_2c_2 + a_2b_2$ $\quad c_3 =$

$c_4 = b_3c_3 + a_3c_3 + a_3b_3$ $\quad c_4 =$

$c_0 = \phi$

$(b_1 + a_1)(b_0c_0 + a_0c_0 + a_0b_0) + a_1b_1$





Not feasible! Why?

Carry-lookahead adder

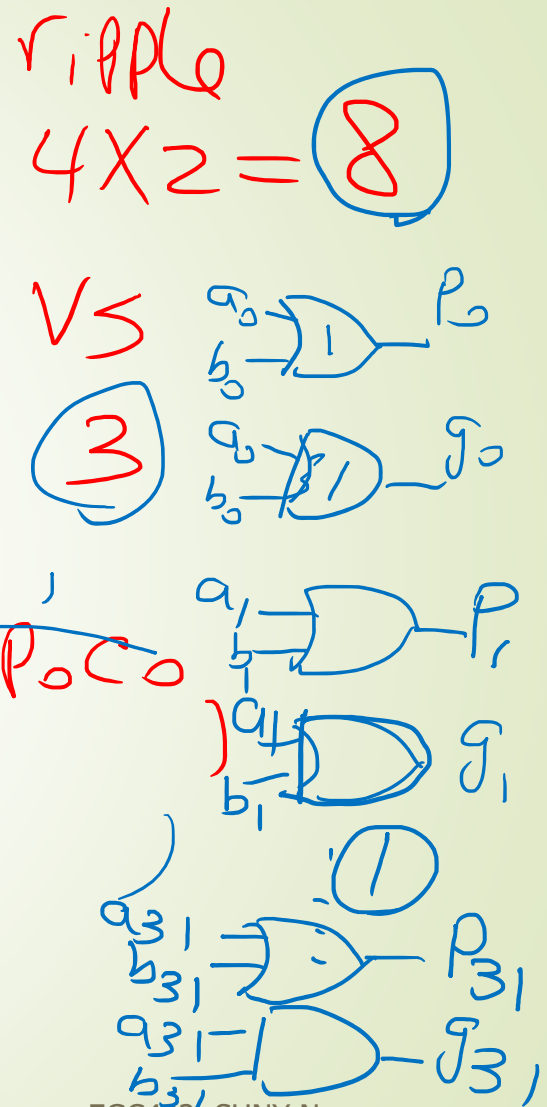
- An approach in-between our two extremes
- $c_1 = b_0c_0 + a_0c_0 + a_0b_0 = (b_0 + a_0)c_0 + a_0b_0$
  - If we didn't know the value of carry-in, what could we do?
  - When would we always generate a carry? $g_i = a_i\, b_i$
  - When would we propagate the carry? $p_i = a_i + b_i$
- Did we get rid of the ripple?

$c_1 = g_0 + p_0c_0$

$c_2 = g_1 + p_1c_1$ $\quad c_2 =$

$c_3 = g_2 + p_2c_2$ $\quad c_3 =$

$c_4 = g_3 + p_3c_3$ $\quad c_4 =$

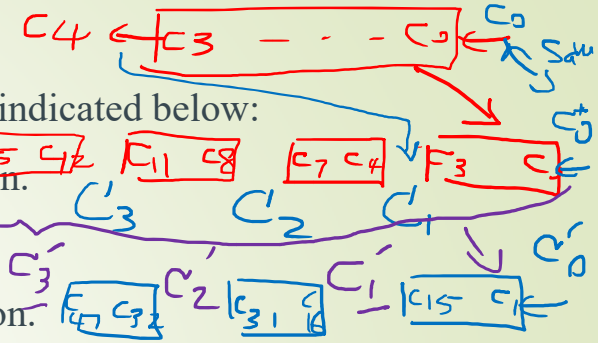


2. Assume you are asked to design a 64 bit carry lookahead carry adder as indicated below:

   a. At the level one, use p_i and g_i and c_i, to express the Boolean function.

   b. At the second level use P_i, G_i, C_i to express the Boolean function.

   c. At the third level use P_i', G_i', and C_i' to express the Boolean function.



$$c_1 = b_0c_0 + a_0c_0 + a_0b_0$$

$(b_0 + a_0)c_0 + a_0b_0 \rightarrow C_1 = P_0G_0 + P_0C_0$

$C_1 = g_0 + P_0C_0$

$C_2 = g_1 + P_1g_0 + P_1P_0C_0$ ✓

$C_3 = g_2 + P_2g_1 + P_2P_1g_0 + P_2P_1P_0C_0$

$C_4 = g_3 + P_3g_2 + P_3P_2g_1 + P_3P_2P_1g_0 + P_3P_2P_1P_0C_0$

$C_2 \rightarrow G_3 = G_2 + P_2G_1 + P_2P_1G_0 + P_2P_1P_0C_0$

$C_{16} \rightarrow C_4 = G_3 + P_3G_2 + P_3P_2G_1 + P_3P_2P_1G_0 + P_3P_2P_1P_0C_0$

$C_1 = G_0 + P_0C_0$

$C_4 = G_1 + P_1G_0 + P_1P_0C_0$

$C_2 = G_1 + P_1G_0 + P_1P_0C_0$

$C_{32} \leftarrow C_2'$ P_0

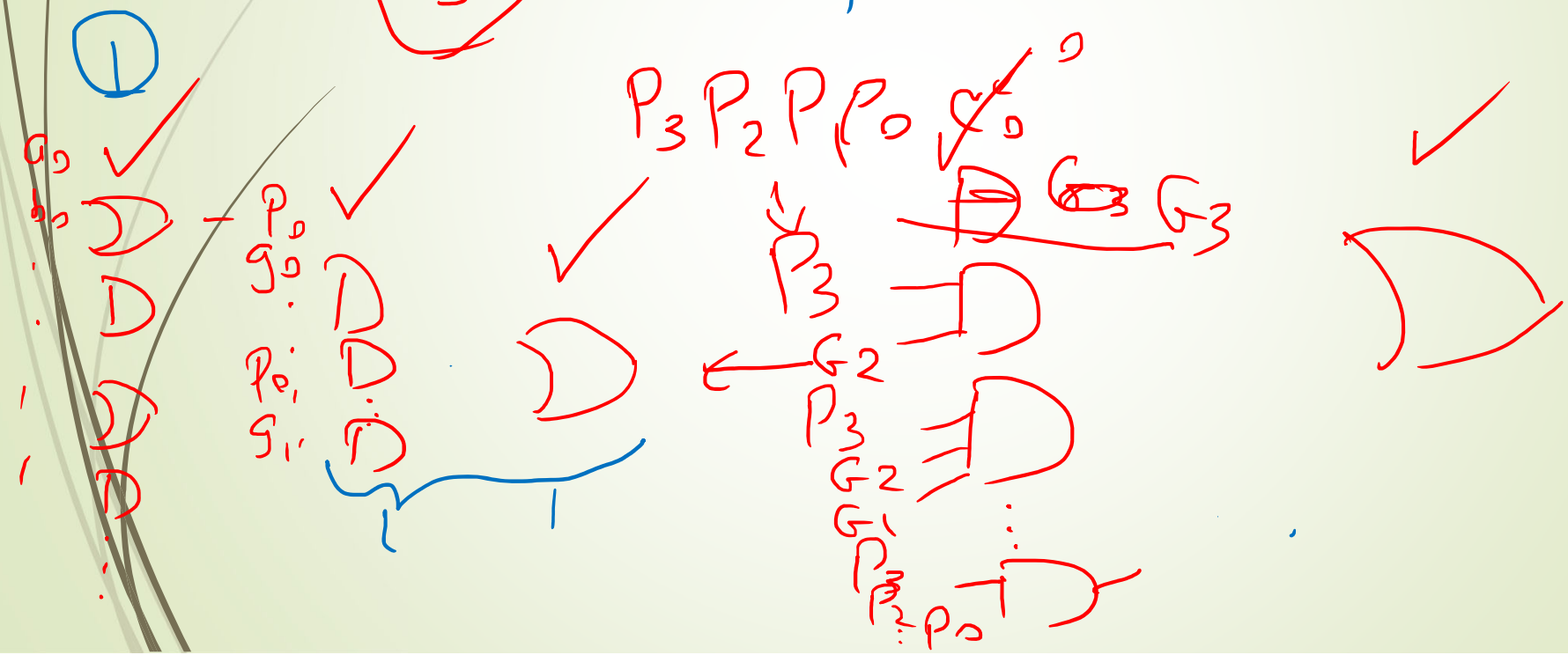
$C_{48} \leftarrow C_3'$

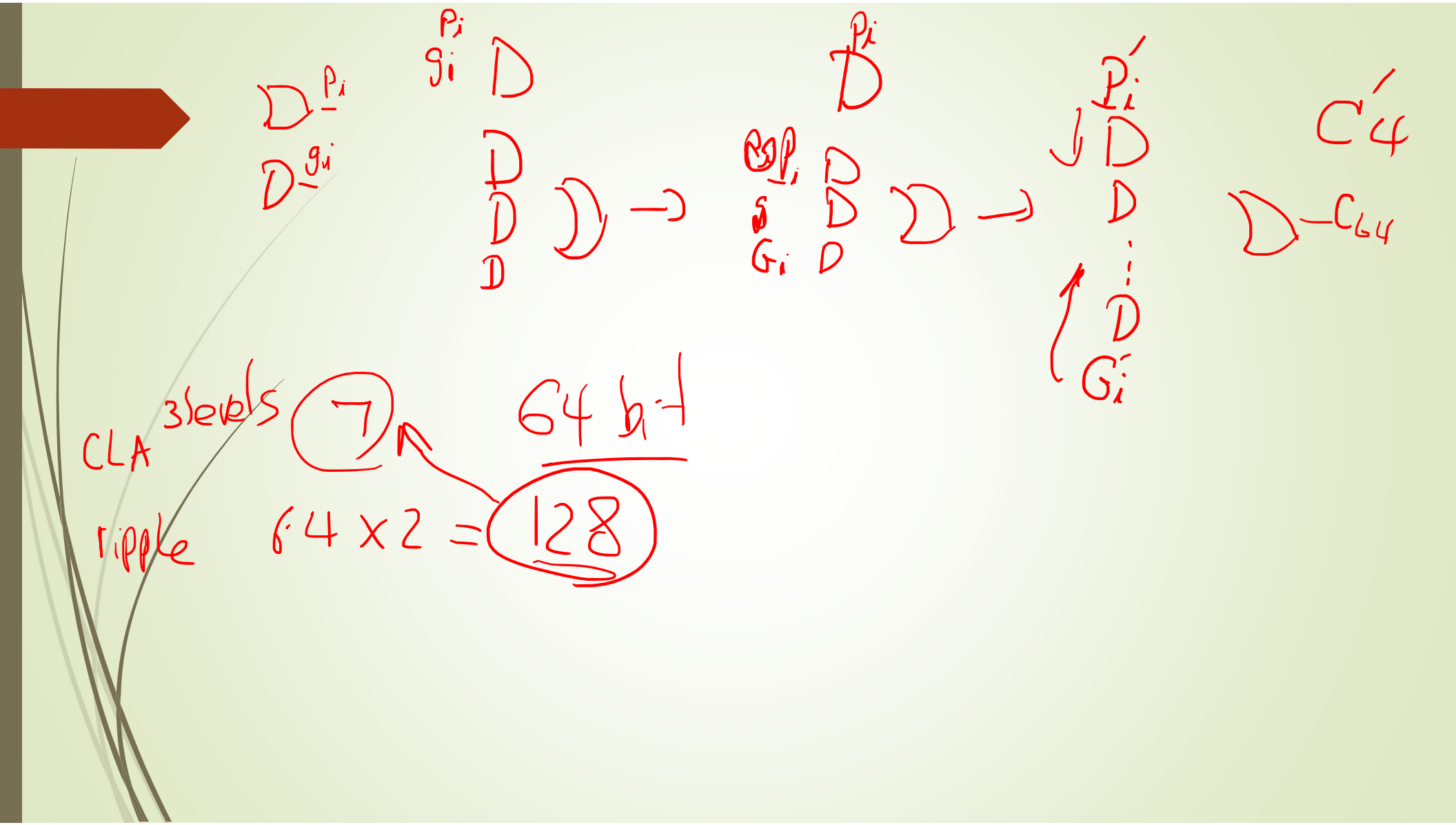
$C_{64} \rightarrow C_4' = G_3' + P_3'G_2' + P_3'P_2'G_1' + P_3'P_2'P_1'G_0' + P_3'P_2'P_1'P_0'C_0$

$\checkmark$ 16 X 2 = 32 Ripple 16 bit

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

$P_3 P_2 P_1 P_0 C_0$



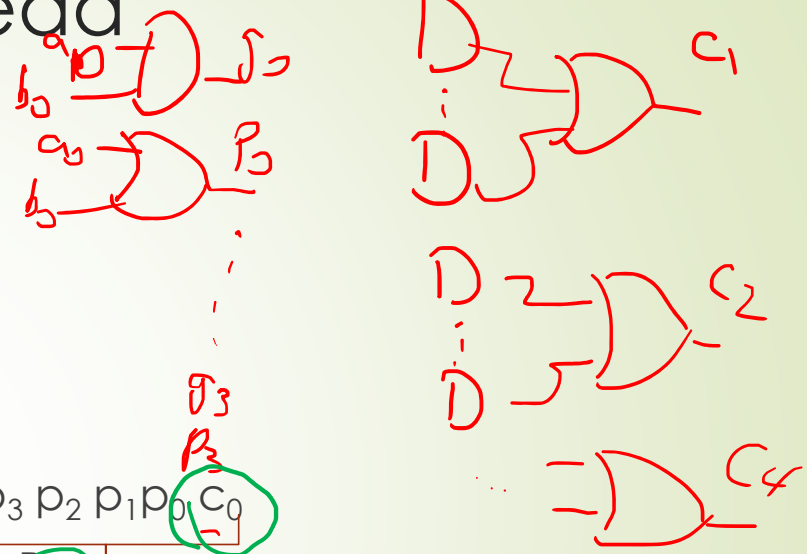

P_i
g_i
D
P_i
G_i
P_i
P_i
C_4
C_{64}
CLA 3 levels
7
64 bit
ripple
64 x 2 = 128

16 Bit Carry Look Ahead

- $g_i = a_i\, b_i$
- $p_i = a_i + b_i$
- $c_1 = g_0 + p_0\, c_0$
- $c_2 = g_1 + p_1\, g_0 + p_1 p_0 c_0$
- $c_3 = g_2 + p_2\, g_1 + p_2\, p_1\, g_0 + p_2\, p_1 p_0\, c_0$
- $c_4 = g_3 + p_3 g_2 + p_3 p_2\, g_1 + p_3\, p_2\, p_1\, g_0 + p_3\, p_2\, p_1 p_0\, c_0$

G_0 P_0

- $G_0 = g_3 + p_3\, g_2 + p_3\, p_2\, g_1 + p_3\, p_2\, p_1\, g_0$ $\quad P_0 = p_3\, p_2\, p_1\, p_0$
- $G_1 = g_7 + p_7\, g_6 + p_7\, p_6\, g_5 + p_7\, p_6\, p_5\, g_4$ $\quad P_1 = p_7\, p_6\, p_5\, p_4$
- $G_2 = g_{11} + p_{11}\, g_{10} + p_{11}\, p_{10}\, g_9 + p_{11}\, p_{10}\, p_9\, g_8$ $\quad P_2 = p_{11}\, p_{10}\, p_9\, p_8$
- $G_3 = g_{15} + p_{15}\, g_{14} + p_{15}\, p_{14}\, g_{13} + p_{15}\, p_{14}\, p_{13}\, g_{12}$ $\quad P_3 = p_{15}\, p_{14}\, p_{13}\, p_{12}$



1. Determine the g_i, p_i, P_i, and G_i values of the following two 16 bit numbers. What is $Cout_{15}$ (C_{16})?

0001 1010 0011 0011

\+ 1110 0101 1110 1011

$p_i = a_i + b_i$

$g_i = a_i\, b_i$

c_i

Repeat Using P_i and G_i

$P_0=$ $P_1=$ $P_2=$ $P_3=$

$G_0 =$

$G_1 =$

$G_2 =$

$G_3 =$

$C_4 =$

$P_0 = p_3\, p_2\, p_1 p_0$

$G0 = g_3 + p_3 g_2 + p_3 p_2\, g_1 + p_3\, p_2\, p_1\, g_0$

3. One simple way to model time for logic is to assume each AND and OR gate takes the same time for a signal to pass through it. Time is estimated by simply counting the number of gates along the longest path through a piece of logic. Compare the number of gate delays for the critical paths of the following 64-bit adders
   a. Ripple carry 128
   b. three-level carry lookahead 7
   c. Carry lookahead at level one, and ripple carry between 4 bit modules 16 x 3 = 48
   d. Carry lookahead at levels one and two, and ripple carry between 16 bit modules. 20