

**Lecture 4:**  
**Carry-Lookahead, Carry-Select, Hybrid Adders**

ECE 645—Computer Arithmetic  
2/19/08

*ECE 645 – Computer Arithmetic*

## Lecture Roadmap

---

- Fast Adders
  - Carry-lookahead adders
  - Carry-select adders
  - Hybrid adders

## Required Reading

---

- B. Parhami, *Computer Arithmetic: Algorithms and Hardware Design*
  - Chapter 6, Carry-Lookahead Adders, Sections 6.1-6.2, pp. 91-96.
  - Chapter 7, Variations in Fast Adders, Section 7.3, Carry-Select Adders, pp. 114-116
- Note errata at:
  - [http://www.ece.ucsb.edu/~parhami/text\\_comp\\_arit.htm#errors](http://www.ece.ucsb.edu/~parhami/text_comp_arit.htm#errors)

---

3



## Solutions to Carry Propagate Problem

---

1. Detect the end of propagation rather than wait for the worst-case time
2. Speed-up propagation via
  - look-ahead
  - carry select, etc.
3. Limit carry propagation to within a small number of bits
4. Eliminate carry propagation through the redundant number representation

---

5



## Basic Signals

**Generate signal:**  $g_i = x_i y_i$

**Propagate signal:**  $p_i = x_i \oplus y_i$

**Anihilate (absorb) signal:**  $a_i = \overline{x_i} \overline{y_i} = \overline{x_i + y_i}$

**Transfer signal:**  $t_i = g_i + p_i = \overline{a_i} = x_i + y_i$

$c_{out} = 1$  given  $c_{in} = 1$

**Carry recurrence**

$$c_{i+1} = g_i + c_i p_i = g_i + c_i t_i$$

7

## Unrolling Carry Recurrence

$$\begin{aligned} c_i &= g_{i-1} + c_{i-1} p_{i-1} = \\ &= g_{i-1} + (g_{i-2} + c_{i-2} p_{i-2}) p_{i-1} = g_{i-1} + g_{i-2} p_{i-1} + c_{i-2} p_{i-2} p_{i-1} = \\ &= g_{i-1} + g_{i-2} p_{i-1} + (g_{i-3} + c_{i-3} p_{i-3}) p_{i-2} p_{i-1} = \\ &= g_{i-1} + g_{i-2} p_{i-1} + g_{i-3} p_{i-2} p_{i-1} + c_{i-3} p_{i-3} p_{i-2} p_{i-1} = \\ &= \dots = \\ &= g_{i-1} + g_{i-2} p_{i-1} + g_{i-3} p_{i-2} p_{i-1} + g_{i-4} p_{i-3} p_{i-2} p_{i-1} + \dots + \\ &\quad + g_0 p_1 p_2 \dots p_{i-2} p_{i-1} + c_0 p_0 p_1 p_2 \dots p_{i-2} p_{i-1} = \end{aligned}$$

$$= g_{i-1} + \sum_{k=0}^{i-2} g_k \prod_{j=k+1}^{i-1} p_j + c_0 \prod_{j=0}^{i-1} p_j$$

8

## 4-bit Carry-Lookahead Adder (1)

---

$$c_4 = g_3 + g_2 p_3 + g_1 p_2 p_3 + g_0 p_1 p_2 p_3 + c_0 p_0 p_1 p_2 p_3$$

$$c_3 = g_2 + g_1 p_2 + g_0 p_1 p_2 + c_0 p_0 p_1 p_2$$

$$c_2 = g_1 + g_0 p_1 + c_0 p_0 p_1$$

$$c_1 = g_0 + c_0 p_0$$

---

$$s_0 = x_0 \oplus y_0 \oplus c_0 = p_0 \oplus c_0$$

$$s_1 = p_1 \oplus c_1$$

$$s_2 = p_2 \oplus c_2$$

$$s_3 = p_3 \oplus c_3$$

---

9

## 4-bit Carry-Lookahead Adder (2)

---

$$c_4 = g_3 + c_3 p_3$$

**3 gates less**

$$c_3 = g_2 + g_1 p_2 + g_0 p_1 p_2 + c_0 p_0 p_1 p_2$$

$$c_2 = g_1 + g_0 p_1 + c_0 p_0 p_1$$

$$c_1 = g_0 + c_0 p_0$$

---

$$s_0 = x_0 \oplus y_0 \oplus c_0 = p_0 \oplus c_0$$

$$s_1 = p_1 \oplus c_1$$

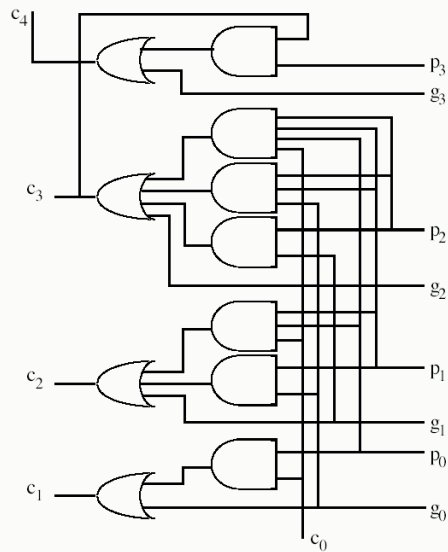
$$s_2 = p_2 \oplus c_2$$

$$s_3 = p_3 \oplus c_3$$

---

10

## 4-bit Carry Network with Full Lookahead



11

## 4-bit Lookahead Carry Generator Equations

$$c_{i+3} = g_{i+2} + g_{i+1} p_{i+2} + g_i p_{i+1} p_{i+2} + c_i p_i p_{i+1} p_{i+2}$$

$$c_{i+2} = g_{i+1} + g_i p_{i+1} + c_i p_i p_{i+1}$$

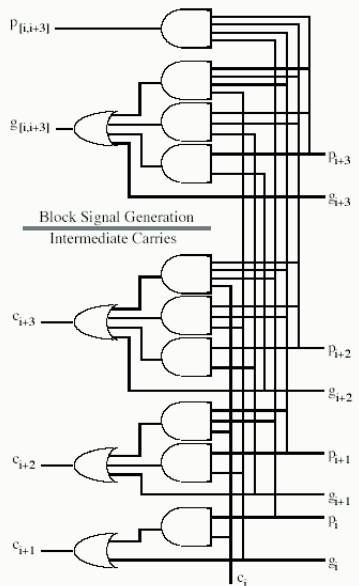
$$c_{i+1} = g_i + c_i p_i$$

$$g_{[i..i+3]} = g_{i+3} + g_{i+2} p_{i+3} + g_{i+1} p_{i+2} p_{i+3} + g_i p_{i+1} p_{i+2} p_{i+3}$$

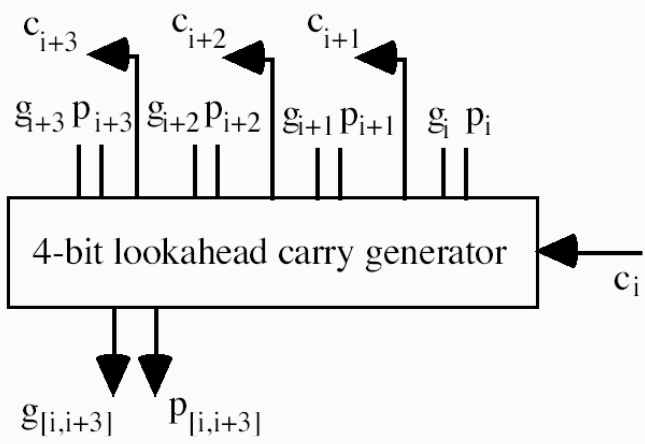
$$p_{[i..i+3]} = p_i p_{i+1} p_{i+2} p_{i+3}$$

12

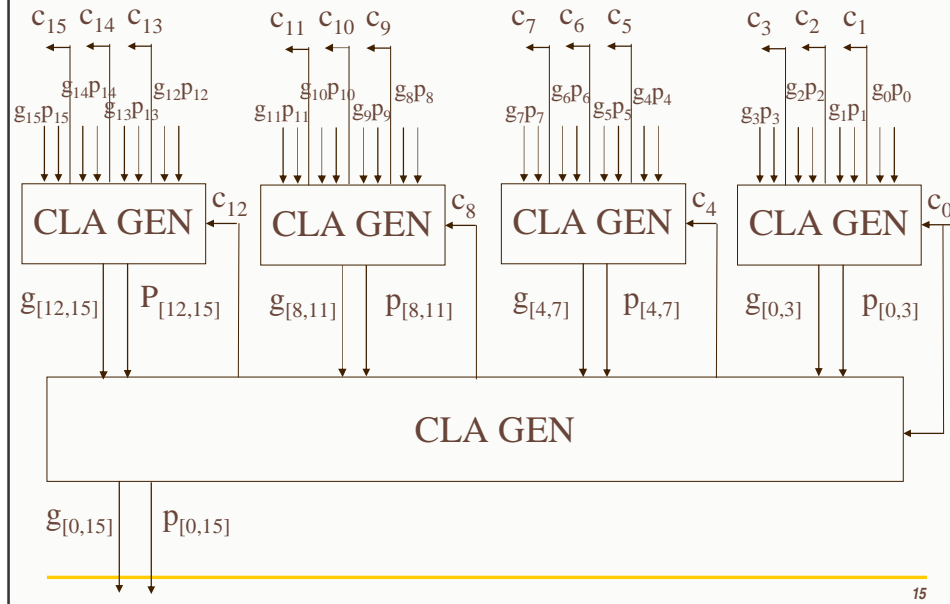
## 4-bit Lookahead Carry Generator Schematic



## 4-bit Lookahead Carry Generator Schematic



## 16-bit 2-level Carry Lookahead Adder



15

## Operation of the 16-bit 2-level Carry Lookahead Adder (1)

Signals computed	Formulas	Delay
$g_i, P_i$ $i=0..15$	$g_i = x_i y_i$ $p_i = x_i \oplus y_i$	1 gate delay
$g_{[i..i+3]}, P_{[i..i+3]}$ $i=0, 4, 8, 12$	$g_{[i..i+3]} = g_{i+3} + g_{i+2} p_{i+3} + g_{i+1} p_{i+2} p_{i+3} + g_i p_{i+1} p_{i+2} p_{i+3}$ $P_{[i..i+3]} = p_i p_{i+1} p_{i+2} p_{i+3}$	2 gate delays

16

## Operation of the 16-bit 2-level Carry Lookahead Adder (2)

Signals computed	Formulas	Delay
$c_4, c_8, c_{12}$ $g_{[0..15]}, P_{[0..15]}$		2 gate delays
	$c_4 = g_{[0..3]} + c_0 P_{[0..3]} \quad c_8 = g_{[4..7]} + g_{[0..3]} P_{[4..7]} + c_0 P_{[0..3]} P_{[4..7]}$ $c_{12} = g_{[8..11]} + g_{[4..7]} P_{[8..11]} + g_{[0..3]} P_{[4..7]} P_{[8..11]} + c_0 P_{[0..3]} P_{[4..7]} P_{[8..11]}$ $g_{[0..15]} = g_{[12..15]} + g_{[8..11]} P_{[12..15]} + g_{[4..7]} P_{[8..11]} P_{[12..15]} + g_{[0..3]} P_{[4..7]} P_{[8..11]} P_{[12..15]}$ $P_{[0..15]} = P_{[0..3]} P_{[4..7]} P_{[8..11]} P_{[12..15]}$	

17

## Operation of the 16-bit 2-level Carry Lookahead Adder (3)

Signals computed	Formulas	Delay
$c_{i+1}, c_{i+2}, c_{i+3}$ $i = 4, 8, 12$		2 gate delays
<i>i.e.</i> , $c_5, c_6, c_7, c_9, c_{10}, c_{11}, c_{13}, c_{14}, c_{15}$	$c_{i+3} = g_{i+2} + g_{i+1} P_{i+2} + g_i P_{i+1} P_{i+2} + c_i P_i P_{i+1} P_{i+2}$ $c_{i+2} = g_{i+1} + g_i P_{i+1} + c_i P_i P_{i+1}$ $c_{i+1} = g_i + c_i P_i$	

18

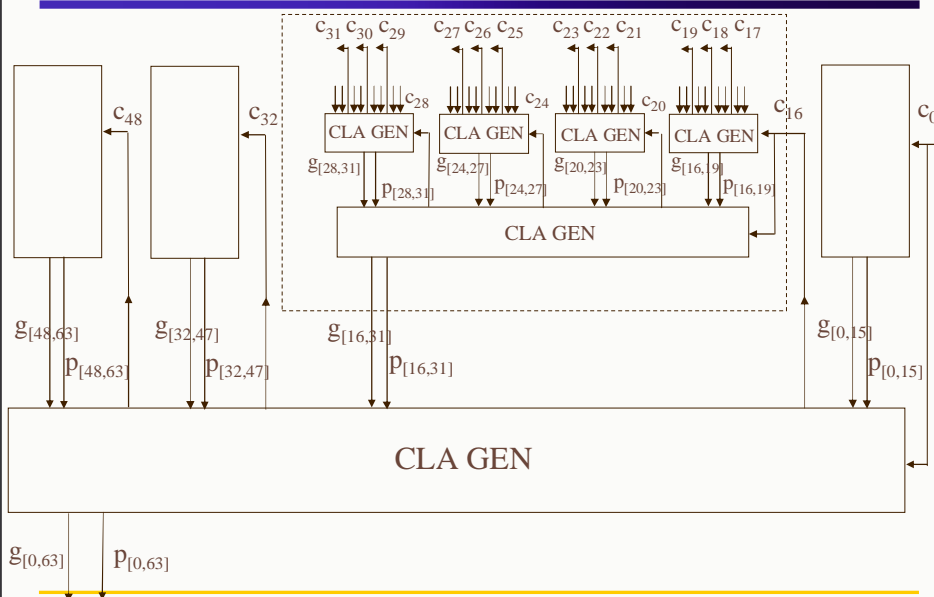
## Operation of the 16-bit 2-level Carry Lookahead Adder (4)

Signals computed	Formulas	Delay
$s_{i+1}, s_{i+2}, s_{i+3}$ $i = 4, 8, 12$ <i>i.e.</i> , $s_5, s_6, s_7, s_9, s_{10}, s_{11}, s_{13}, s_{14}, s_{15}$ $s_i = p_i \oplus c_i$		1 gate delay

**Total: 8 gate levels in the CLA adder vs.  
32 gate levels in the ripple carry adder**

19

## 64-bit 3-level Carry Lookahead Adder



20

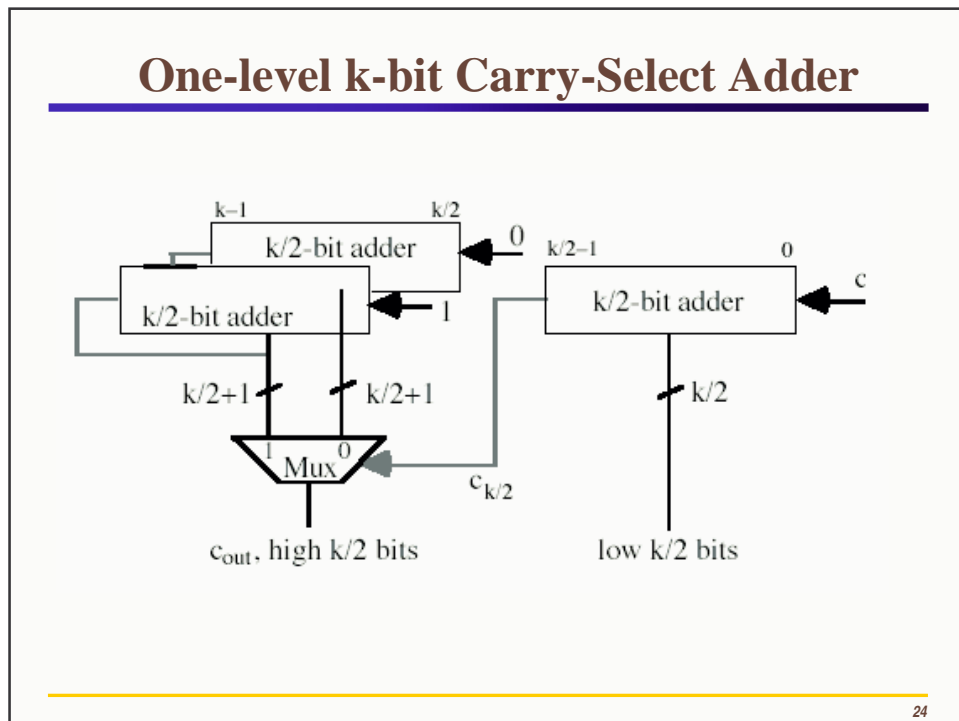
## Operation of the 64-bit 3-level Carry Lookahead Adder

Level	Signals computed	Delay
PRE	$g_i, p_i \quad i=0..63$	1 gate delay
1	$g_{[i..i+3]}, p_{[i..i+3]} \quad i=0, 4, 8, 12, \dots, 56, 60$	2 gate delays
2	$g_{[i..i+15]}, p_{[i..i+15]} \quad i=0, 16, 32, 48$	2 gate delays
3	$g_{[0..63]}, p_{[0..63]} \quad c_{16}, c_{32}, c_{48},$	2 gate delays
2	$c_{20}, c_{24}, c_{28}, c_{36}, c_{40}, c_{44}, c_{52}, c_{56}, c_{60}$	2 gate delays
1	$c_{21}, c_{22}, c_{23}, c_{25}, c_{26}, c_{27}, \dots, c_{61}, c_{62}, c_{63}$	2 gate delays
POST	<del><math>s_{21}, s_{22}, s_{23}, s_{25}, s_{26}, s_{27}, \dots, s_{61}, s_{62}, s_{63}</math></del>	<del>1 gate delay</del>

## Delay of a k-bit Carry-Lookahead Adder

$$T_{\text{lookahead-adder}} = 4 \lceil \log_4 k \rceil$$

k	$T_{\text{lookahead-adder}}$	$T_{\text{ripple-carry-adder}}$
4	4	8
16	8	32
32	12	64
64	12	128
128	16	256
256	16	512



## One-level k-bit Carry-Select Adder Cost & Latency

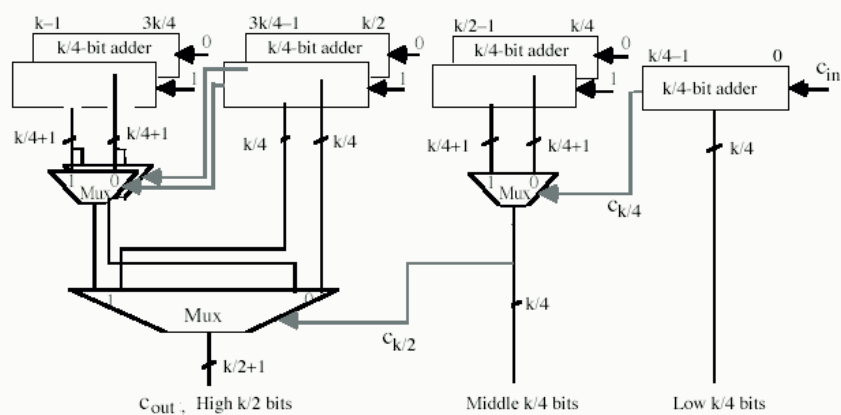
$$C_{\text{select-add}}(k) = 3C_{\text{add}}(k/2) + k/2 + 1$$

$$T_{\text{select-add}}(k) = T_{\text{add}}(k/2) + 1$$

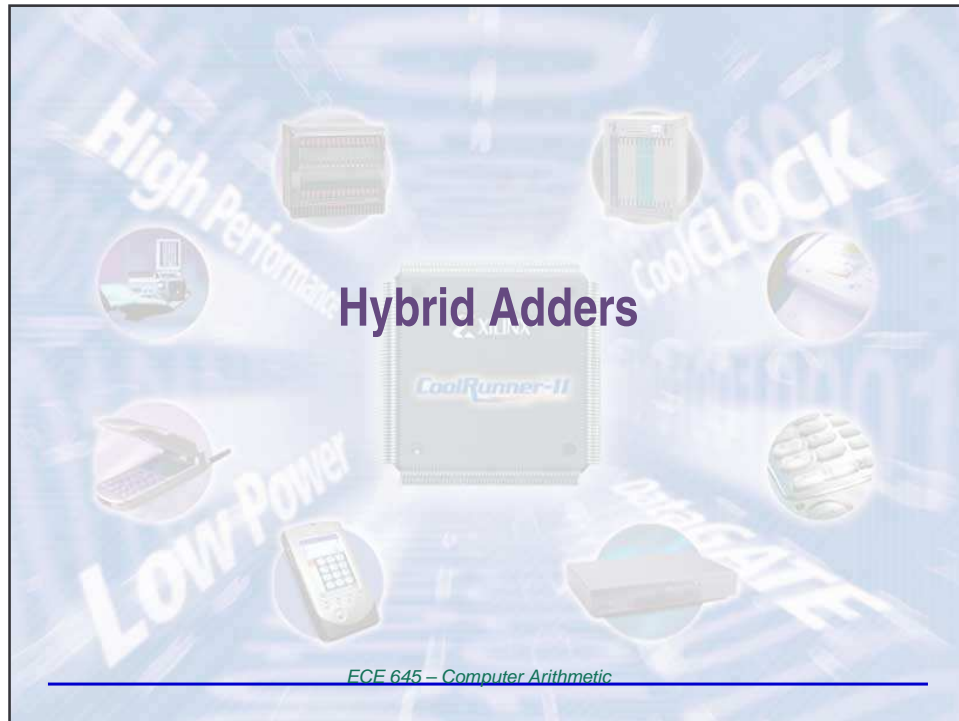
Units: cost and delay of a single 2-to-1 multiplexer

25

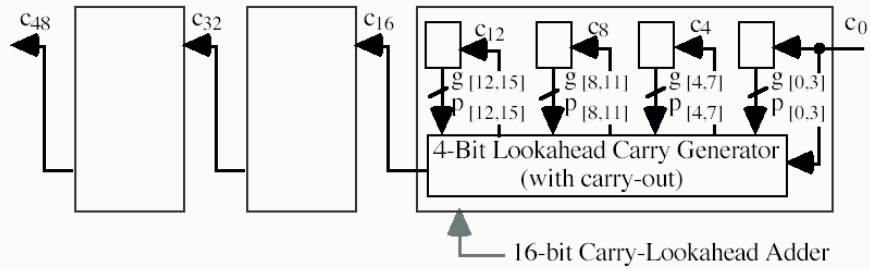
## Two-level k-bit Carry Select Adder



26



### A Hybrid Ripple-Carry/Carry-Lookahead Adder



## A Hybrid Carry-Lookahead/Carry-Select Adder

