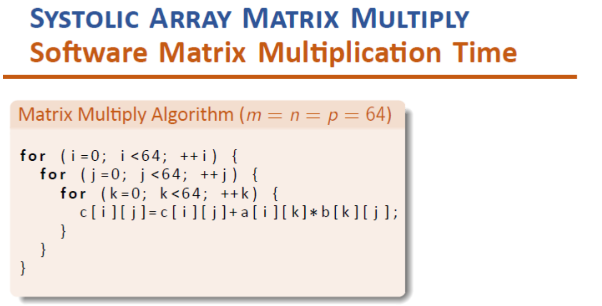
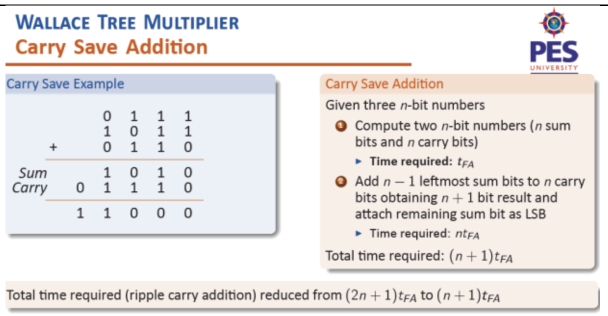
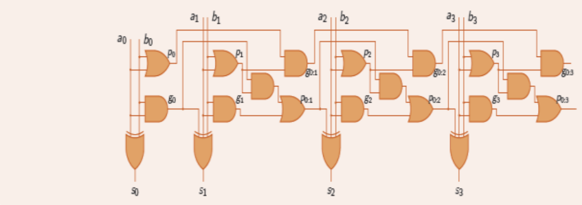
# PES UNIVERSITY
# 3rd SEMESTER

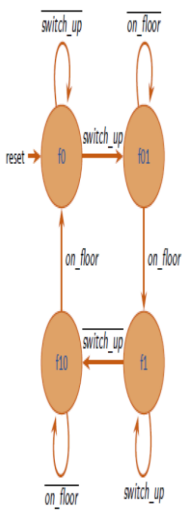
## *Errata for*
## *DIGITAL DESIGN AND COMPUTER ORGANISATION*
## *Slides Uploaded to PESU ACADEMY Portal*

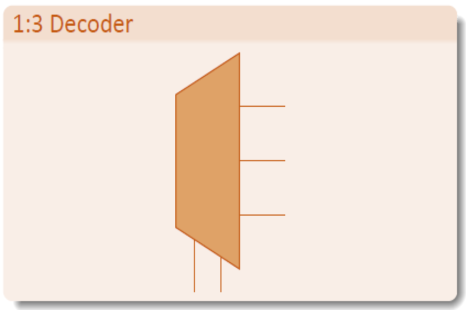
*Errata Last updated 19th November 2020*

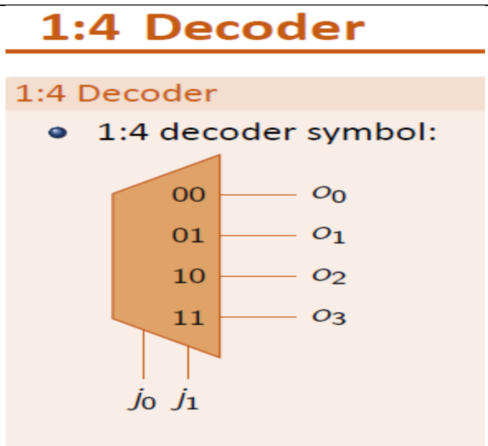
**This list is a work in progress. Some of the following corrections may be revised, and additional corrections will probably be added.**

<table>
<tr><th>Sl. No</th><th>Lecture Number</th><th>Content in the slide</th><th>To be Corrected as</th></tr>
<tr><td>1.</td><td>UNIT5<br>Lecture 53<br>Slide 8</td><td>
</td><td>The total time needed for computation (of the matrix C) with the 2D systolic array will be equal to 3n-2 clock cycles to perform nxn matrix multiplication.</td></tr>
<tr><td>2.</td><td>UNIT4<br>Lecture 38<br>Slide 8</td><td>
</td><td>A tree of adders can be formed, taking only O(lg m · lg n) gate delays.</td></tr>
</table>

<table>
<tr><td>3.</td><td>UNIT3<br>Lecture 34<br>Slide 16</td><td>CARRY-LOOKAHEAD AND PREFIX ADDERS - 4<br>Associative Ripple Carry?<br>Ripple Carry Adder<br><br>• $c_{i+1} = ab + bc_i + c_i a$<br>• Generate and Propagate:<br>▸ $g_i$ carry generated in position $i$<br>▸ $p_i$ carry propagated in position $i$<br>▸ $g_{0:i}$ carry generated in positions 0 to $i$<br>▸ $p_{0:i}$ carry propagated in positions 0 to $i$<br>• $g_i = a_i b_i$<br>• $p_i = a_i + b_i$<br>• $g_{0:i+1} = g_i + p_i g_{0:i}$<br>• $p_{0:i+1} = p_i p_{0:i}$</td><td>In the figure the labels p0: i and g0:1 need to be interchanged.</td></tr>
<tr><td>4.</td><td>UNIT3<br>Lecture 31<br>Slide 5</td><td>• Time requirements: For an $n$-bit ripple carry adder, critical path delay is composed of:<br>▸ Propagation delay from $c_0$ to $c_{n-1}$<br>★ Signal passes through two gates in each of the $n-1$ stages<br>★ $2(n-1)t_g$ time required<br>▸ Sum computation<br>★ $2t_g$ time required for three input XOR gate<br>• An $n$-bit ripple carry adder thus occupies $2nt_g$ time</td><td>Propagation delay from c0 to c(n-1) should be 3(n-1)tg instead of 2(n-1) tg because although the carry signal passes through two gates in each stage, one of the gates has three inputs which counts as two 2 input gates..</td></tr>
<tr><td>5.</td><td>UNIT3<br>Lecture 31<br>Slide 34,35</td><td>CARRY-LOOKAHEAD AND PREFIX ADDERS - 1<br>Performance Comparison<br>PES UNIVERSITY ONLINE<br>• Area and time estimates for $n$-bit adders:<br><br>| | Area | Time |<br>|---|---|---|<br>| Ripple carry | $7na_g$ | $2nt_g$ |<br>| Carry-lookahead | $(n^2+5n)a_g$ | $2\lceil \log_2(n-1) \rceil t_g + 3t_g$ |<br><br>• Compared to the ripple carry adder's linear delay increase with size, the carry-lookahead adder delay increase only logarithmically, resulting in dramatically faster adders<br>• However, the area of the carry-lookahead adder increase qudratically with size<br>• Is there an adder design that retains the carry-lookahead adder's speed but has significantly lesser area?</td><td>log2(n-1) should be replaced by log2(n).</td></tr>
</table>

<table>
<tr>
<td>6.</td>
<td>UNIT3<br>Lecture 24<br>Slide 42 till<br>Slide 83</td>
<td>


Elevator Example State Tansition Table
<table>
<tr><th colspan="4">Current State</th><th colspan="2">Inputs</th><th colspan="4">Next State</th></tr>
<tr><th>$s_3$</th><th>$s_2$</th><th>$s_1$</th><th>$s_0$</th><th>switch_up</th><th>on_floor</th><th>$s_3'$</th><th>$s_2'$</th><th>$s_1'$</th><th>$s_0'$</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td></tr>
</table>
</td>
<td>on_first should be equal to 1 in state 3(0100) instead of the incorrectly mentioned on_first should be equal to 1 in state 4(1000)<br>'lift_down' be 1 in state 4 (1000) according to the state diagram</td>
</tr>
<tr>
<td>7.</td>
<td>UNIT 2,<br>Lecture 14<br>Slide 20</td>
<td>


</td>
<td>To be corrected as 2:3 decoder</td>
</tr>
<tr>
<td>8.</td>
<td>UNIT 2<br>Lecture 14<br>Slide<br>12-16</td>
<td>


</td>
<td>To be corrected as 2:4 decoder</td>
</tr>
</table>

<table>
<tr>
<td>9.</td>
<td>UNIT1,<br>Lecture 3,<br>Slide 12</td>
<td>
<table>
<tr><th>a</th><th>b</th><th>c</th><th>y</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td></tr>
</table>
Truth table
</td>
<td>
<table>
<tr><th>a</th><th>b</th><th>c</th><th>Y</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td></tr>
</table>
</td>
</tr>
<tr>
<td>10.</td>
<td>UNIT1,<br>Lecture 4,<br>Slide 13</td>
<td>
Boolean Algebra<br>
1. **Set** {0, 1}<br>
2. **Operations** AND, OR, NOT<br>
3. **Identity elements** 0 (for AND), 1 (for OR)<br>
4. **Laws/Identities** Commutative, associative, distributive, . . .
</td>
<td>
**Set** f0; 1g<br>
**Operations** AND, OR, NOT<br>
**Identity elements 1** (for AND), 0 (for OR)<br>
**Laws/Identities** Commutative, associative, distributive:
</td>
</tr>
<tr>
<td>11.</td>
<td>UNIT1,<br>Lecture 8<br>Slide 4 till<br>Slide<br>16,Also<br>same error<br>in Slide 40</td>
<td>
<table>
<tr><th>a</th><th>b</th><th>c</th><th>d</th><th>y</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr>
</table>
Four Input Truth Table
</td>
<td>
<table>
<tr><th>a</th><th>b</th><th>c</th><th>d</th><th>y</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr>
</table>
The min terms for table are F(a ,b, c, d)=Σ(0,2,5,7,8,10,13,15)
</td>
</tr>
</table>