EECS150 - Digital Design

Lecture 8 - Multipliers (part 1): Design Example

Feb 14, 2012

John Wawrzynek

Multiplication

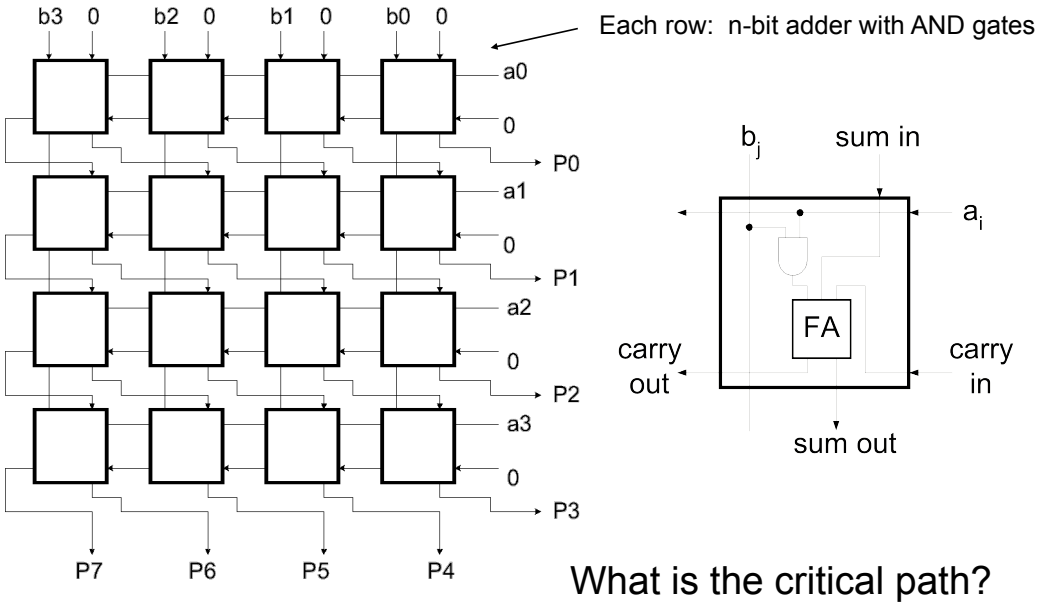
$$
\begin{array}{llllll}
 & & a_3 & a_2 & a_1 & a_0 \quad \leftarrow \textit{Multiplicand} \\
 & & b_3 & b_2 & b_1 & b_0 \quad \leftarrow \textit{Multiplier} \\
\hline
 & X & a_3b_0 & a_2b_0 & a_1b_0 & a_0b_0 \\
 & a_3b_1 & a_2b_1 & a_1b_1 & a_0b_1 & \\
a_3b_2 & a_2b_2 & a_1b_2 & a_0b_2 & & \\
a_3b_3 & a_2b_3 & a_1b_3 & a_0b_3 & & \\
\hline
 & \ldots & a_1b_0+a_0b_1 & a_0b_0 & & \leftarrow \textit{Product}
\end{array}
$$

Partial products

Many different circuits exist for multiplication. Each one has a different balance between speed (performance) and amount of logic (cost).

Array Multiplier

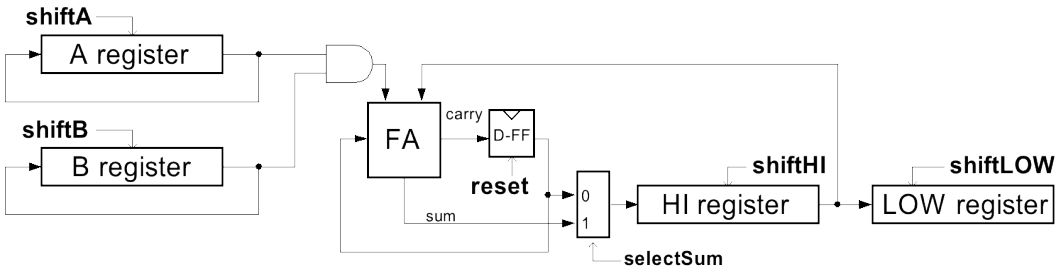
Single cycle multiply: Generates all n partial products simultaneously.



What is the critical path?

Bit-Serial Multiplier

- Example, Bit-serial multiplier (n^2 cycles, one bit of result per n cycles):



- Control Algorithm:

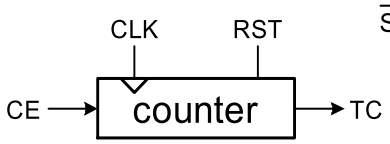
```
repeat n cycles {   // outer (i) loop
        repeat n cycles{     // inner (j) loop
                shiftA, selectSum, shiftHI
        }
        shiftB, shiftHI, shiftLOW, reset
}
```

Note: The occurrence of a control signal x means x=1. The absence of x means x=0.

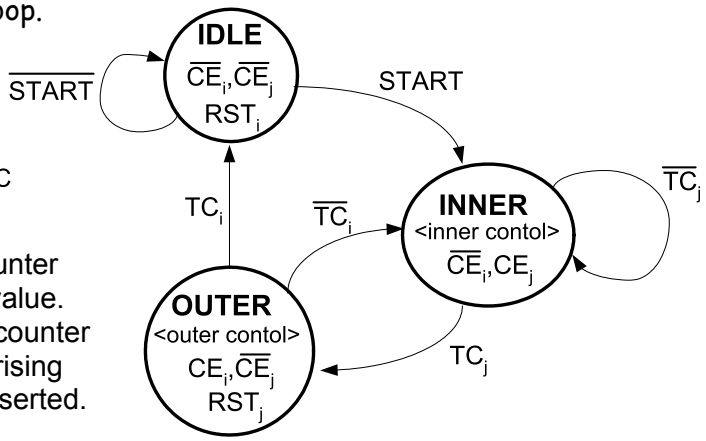
Controller using Counters

- **State Transition Diagram:**
  - Assume presence of two binary counters. An "i" counter for the outer loop and "j" counter for inner loop.

```
repeat n cycles { // outer (i) loop
        repeat n cycles{   // inner (j) loop
                shiftA, selectSum, shiftHI
        }
        shiftB, shiftHI, shiftLOW, reset
}_
```

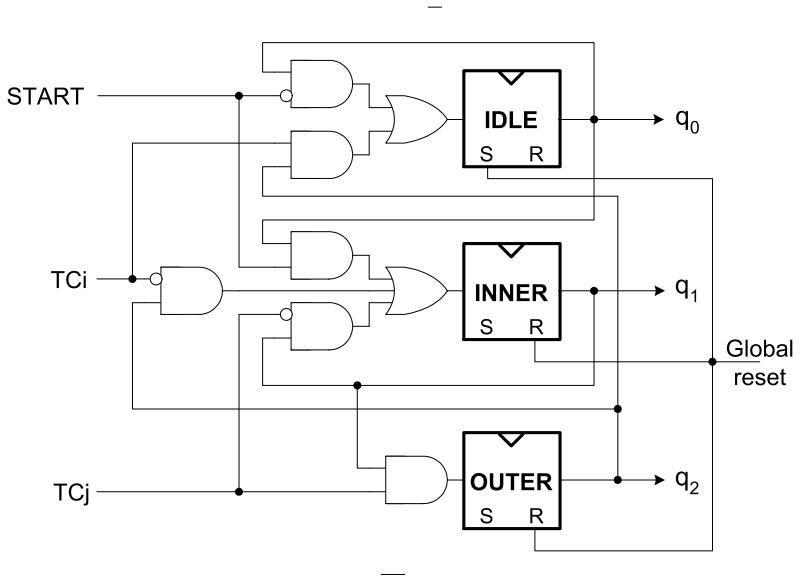


TC is asserted when the counter reaches it maximum count value. **CE** is "count enable". The counter increments its value on the rising edge of the clock if CE is asserted.



Controller using Counters

- **Controller circuit implementation:**



- Outputs:

$CE_i = q_2$
$CE_j = q_1$
$RST_i = q_0$
$RST_j = q_2$

$shiftA = q_1$
$shiftB = q_2$
$shiftLOW = q_2$
$shiftHI = q_1 + q_2$
$reset = q_2$
$selectSUM = q_1$