Introduction to Digital Systems

Part III (Sequential Components)

2021/2022

Iterative vs. Sequential Circuits

Controlpath / Datapath Decomposition of Computational Systems

Arnaldo Oliveira, Augusto Silva, Iouliia Skliarova

Universidade de Aveiro

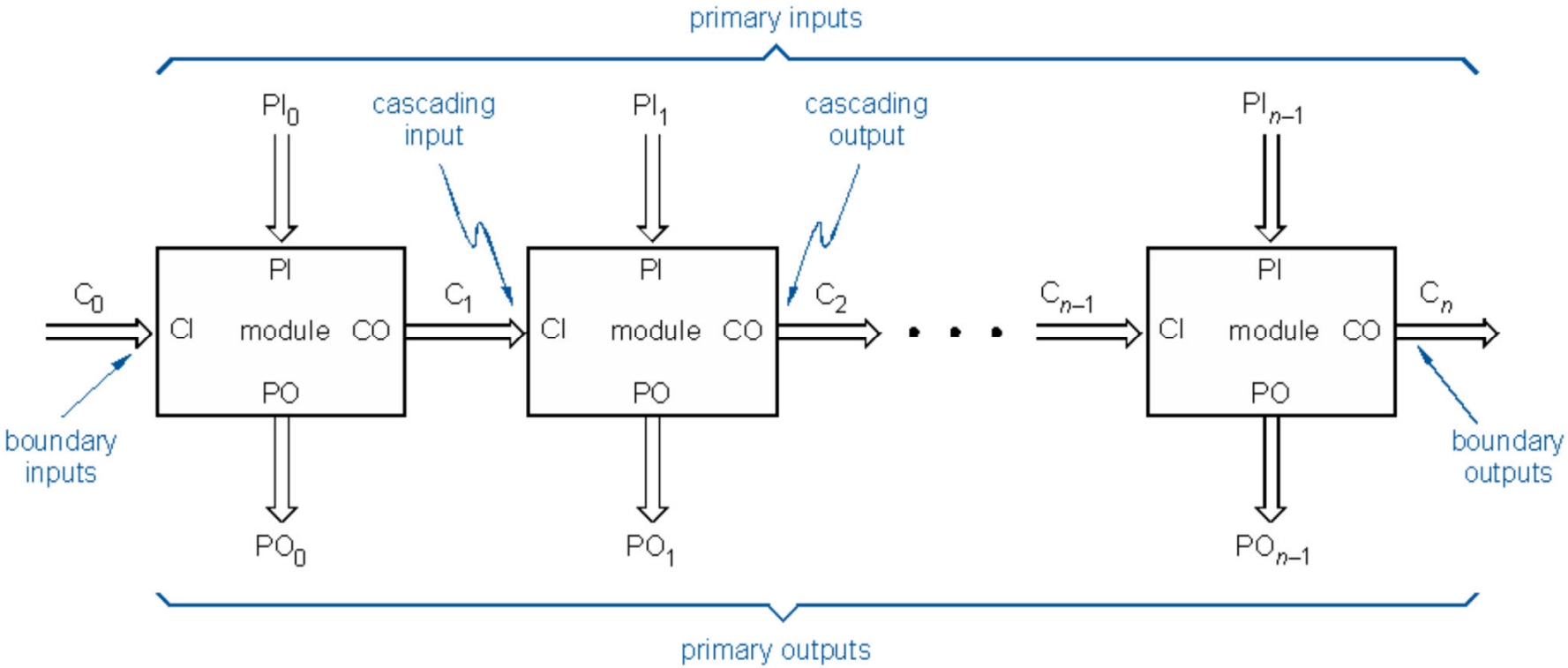
Lecture Contents

- Iterative vs. Sequential circuits
  - Concepts
  - Examples
    - Adder
    - Comparator
    - Unsigned multiplier
- Controlpath – Datapath decomposition of computational systems

Some figures and content extracted from: John F. Wakerly, "Digital Design – Principles and Practices", 4 ed., Pearson – Prentice Hall, 2006 (chapter 7). Reading chapter 8 (4th ed.) or chapter 11 (5th ed.) is highly recommended.

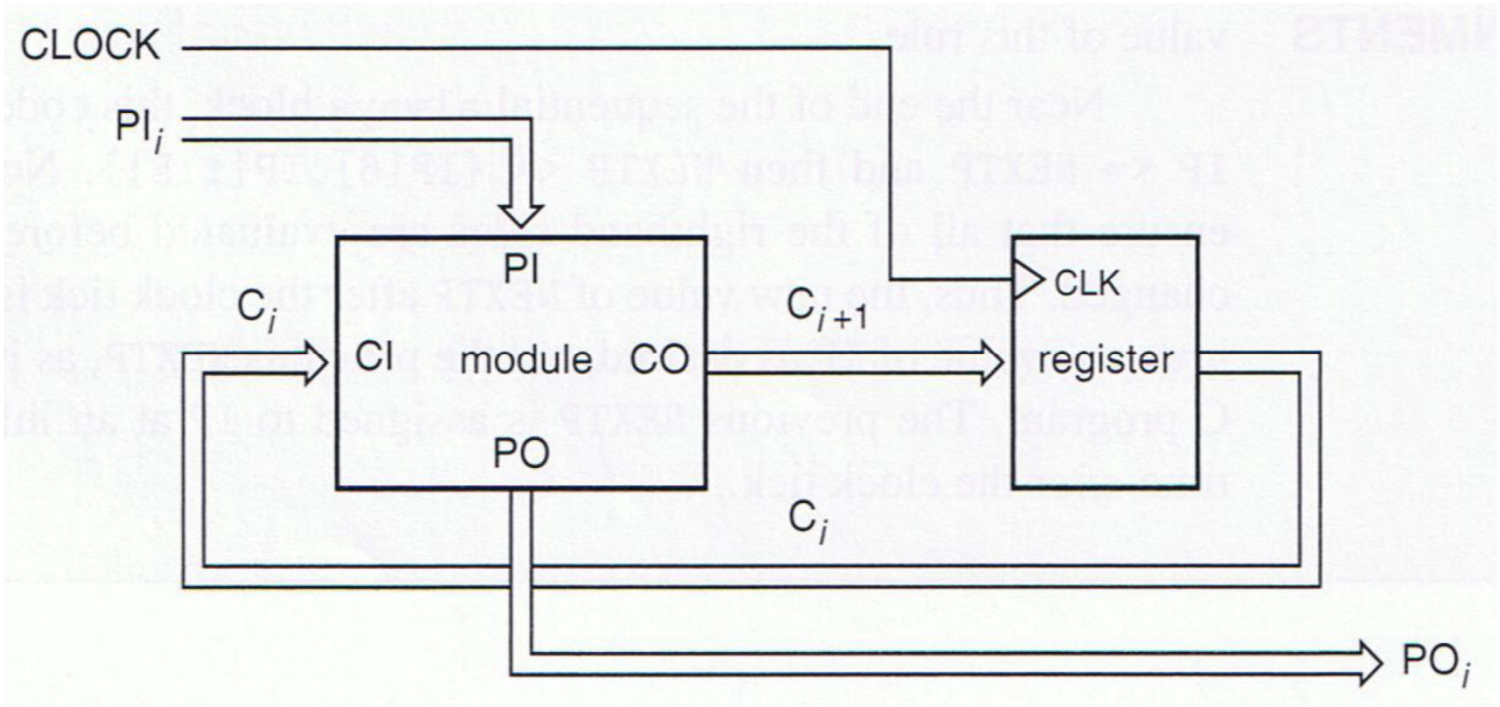
Sequential vs. Iterative Circuits

Iterative circuit (built with several identical processing tiles/building blocks, typically combinatorial with parallel data input and output)



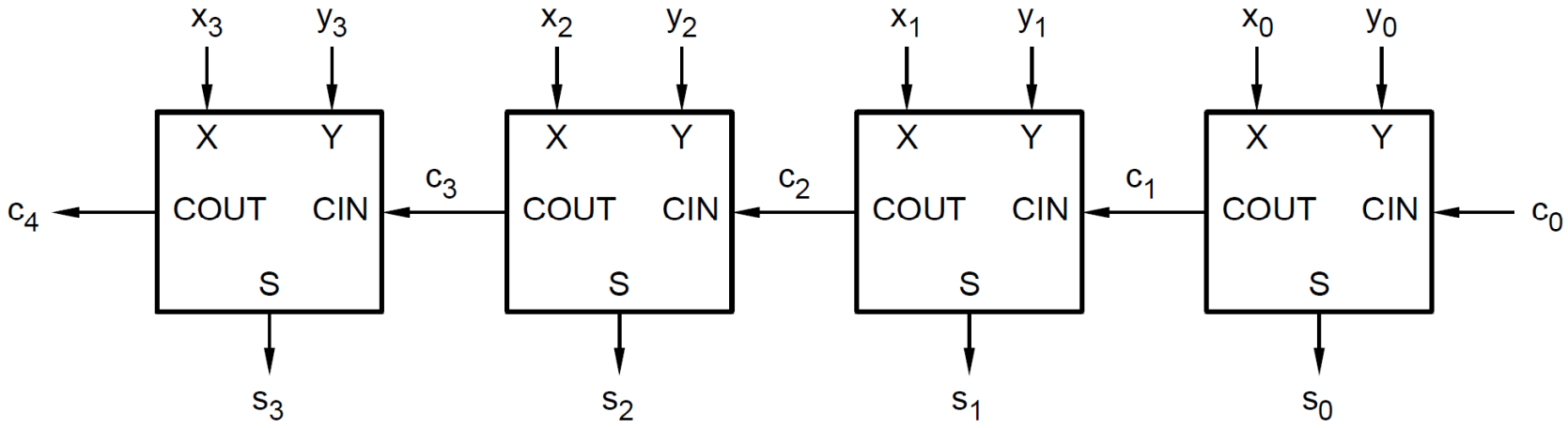
Sequential vs. Iterative Circuits

Sequential circuit (serial data input and output with smaller set of processing tiles/building blocks and a memory element/register)

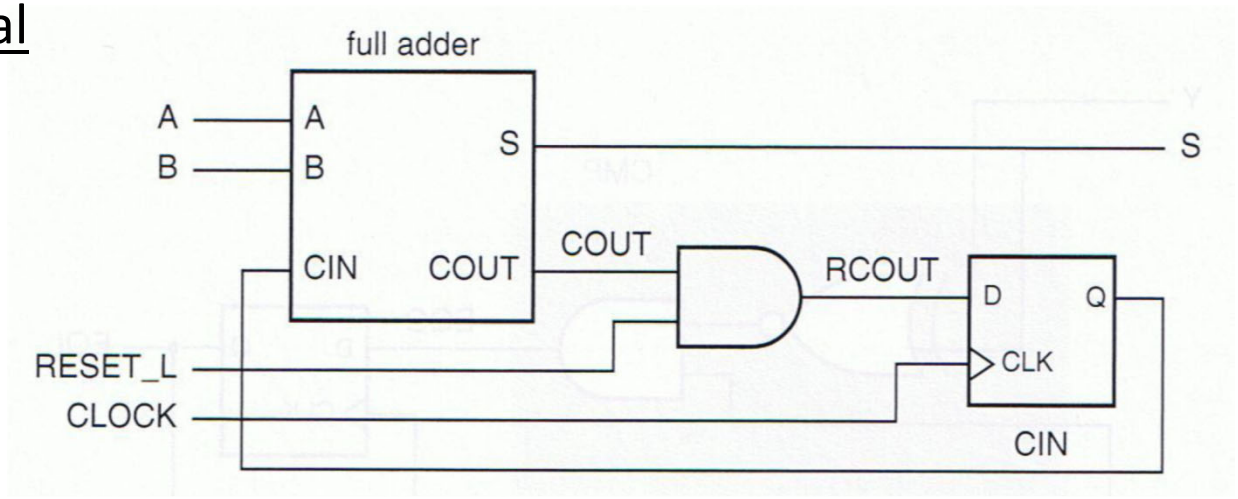


Iterative vs. Sequential Adder

Iterative / Combinatorial

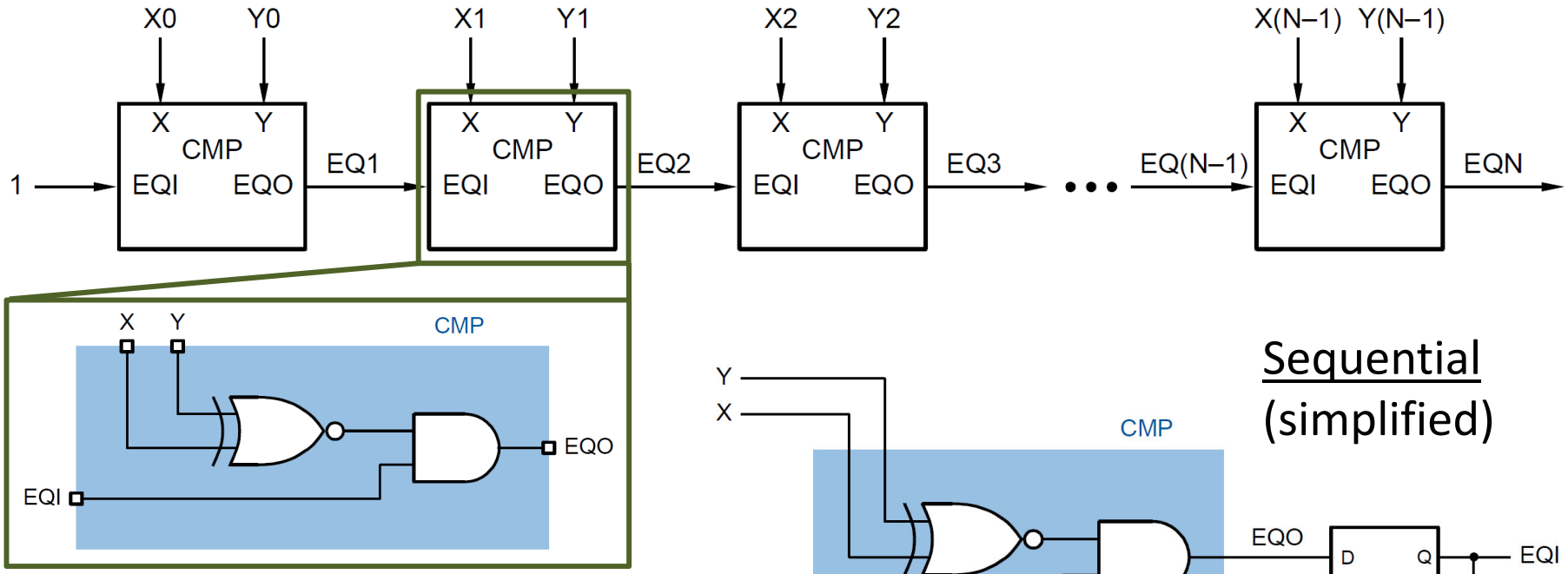


Sequential



Iterative vs. Sequential Comparator

Iterative / Combinatorial



Sequential (simplified)

Sequential Comparator

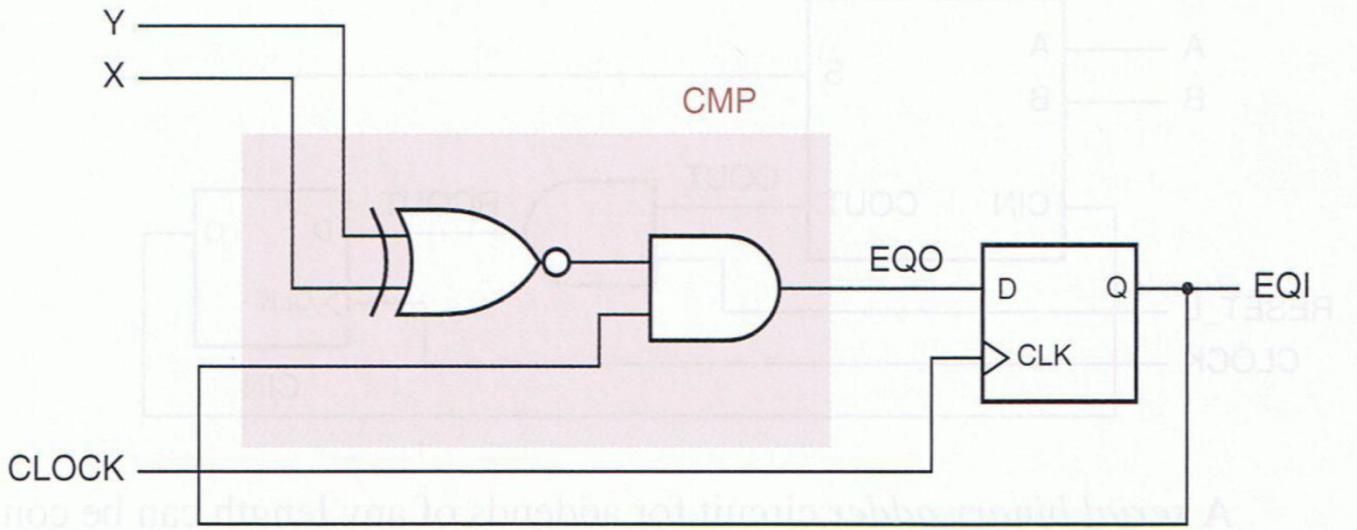


Figure 8-59
Simplified serial comparator circuit.

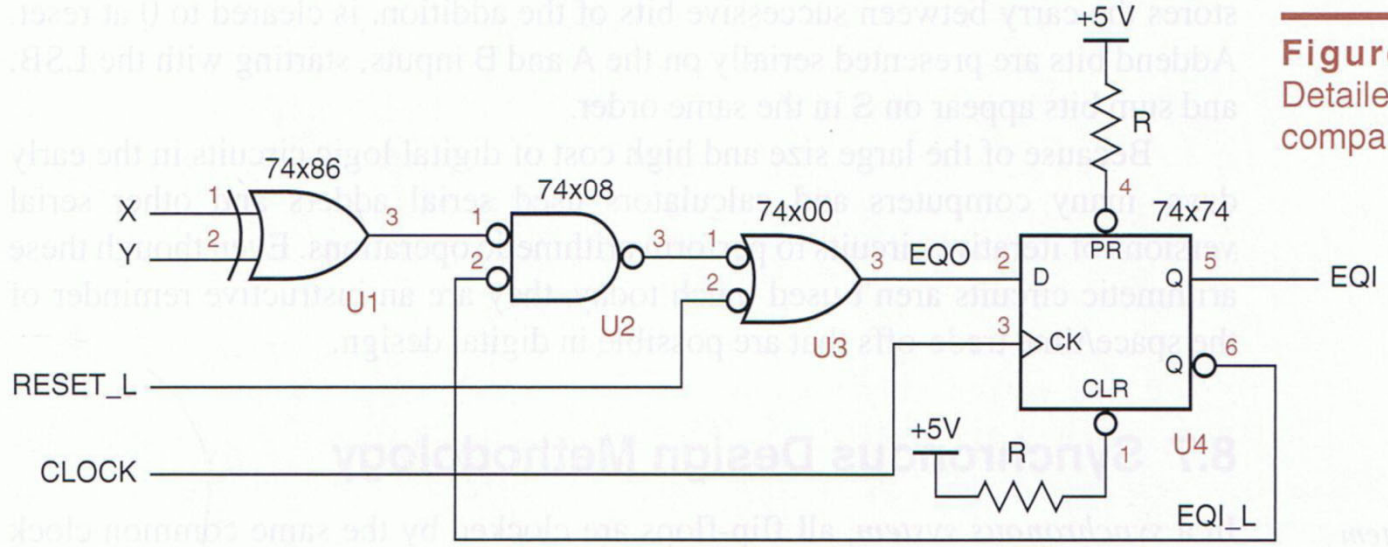


Figure 8-60
Detailed serial comparator circuit.

Sequential Comparator

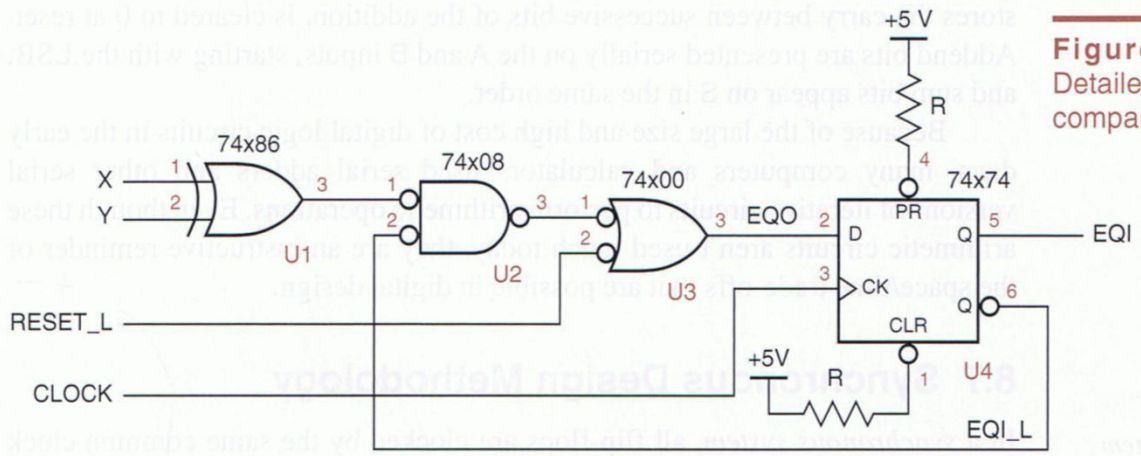


Figure 8-60 Detailed serial comparator circuit.

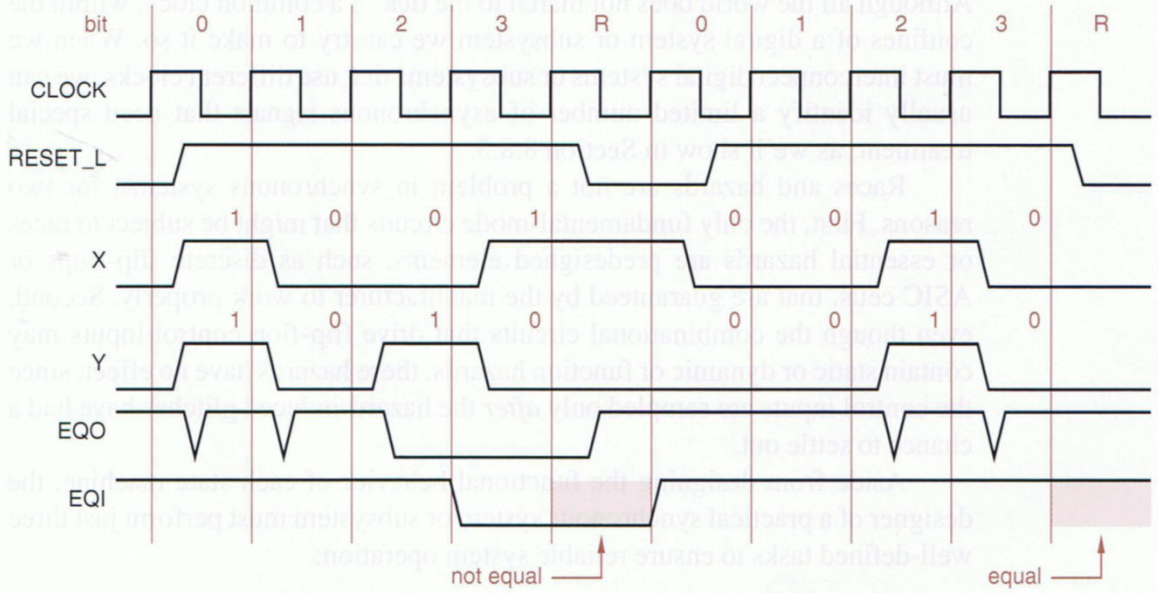


Figure 8-61 Timing diagram for serial comparator circuit.

Unsigned Multiplication

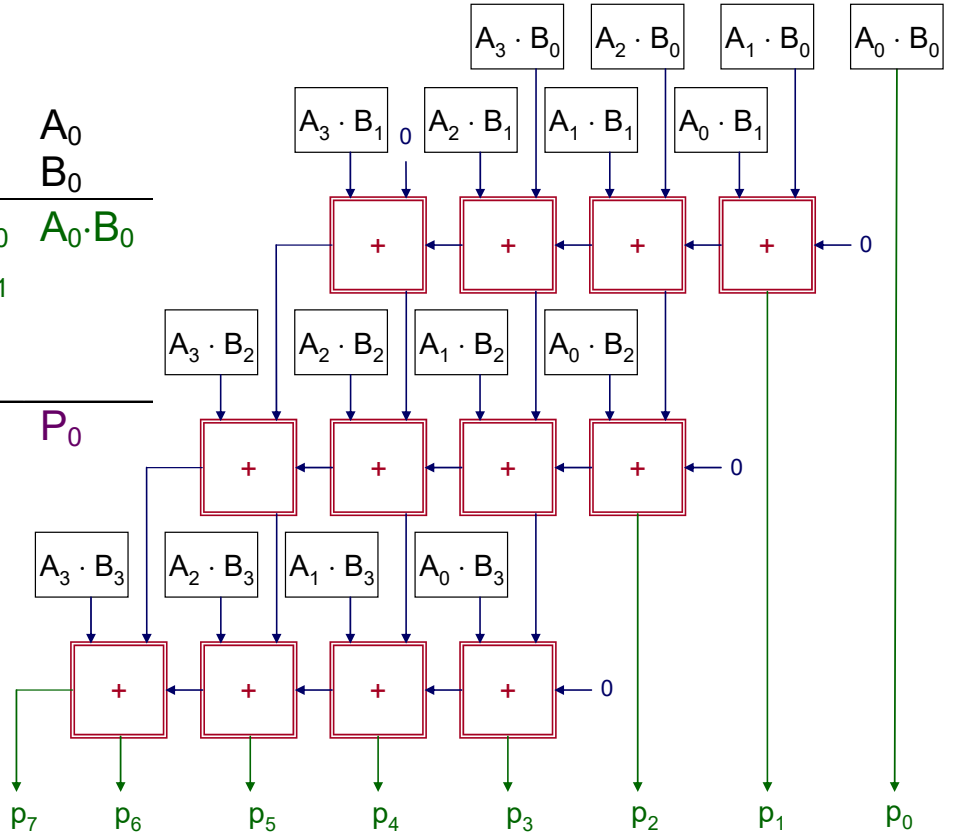
- Mutiplier architecture highly based on the **paper and pencil** algorithm
- Multiplication of two **N bits operands** requires a storage space of **2N bits** for the **result**

$$
\begin{array}{cccccccc}
 & & & & 1 & 1 & 0 & 0 \\
 & & & \times & 1 & 1 & 0 & 1 \\
\hline
 & & & & 1 & 1 & 0 & 0 \\
 & & & 0 & 0 & 0 & 0 & \\
 & & 1 & 1 & 0 & 0 & & \\
 & 1 & 1 & 0 & 0 & & & \\
\hline
1 & 0 & 0 & 1 & 1 & 1 & 0 & 0
\end{array}
$$

Combinational/Iterative/Parallel Unsigned Multiplier

4 bit example

				A_3	A_2	A_1	A_0
			$\times$	B_3	B_2	B_1	B_0
				$A_3 \cdot B_0$	$A_2 \cdot B_0$	$A_1 \cdot B_0$	$A_0 \cdot B_0$
			$A_3 \cdot B_1$	$A_2 \cdot B_1$	$A_1 \cdot B_1$	$A_0 \cdot B_1$	
		$A_3 \cdot B_2$	$A_2 \cdot B_2$	$A_1 \cdot B_2$	$A_0 \cdot B_2$		
	$A_3 \cdot B_3$	$A_2 \cdot B_3$	$A_1 \cdot B_3$	$A_0 \cdot B_3$			
P_7	P_6	P_5	P_4	P_3	P_2	P_1	P_0



16 × 2-input AND gates

12 × adders

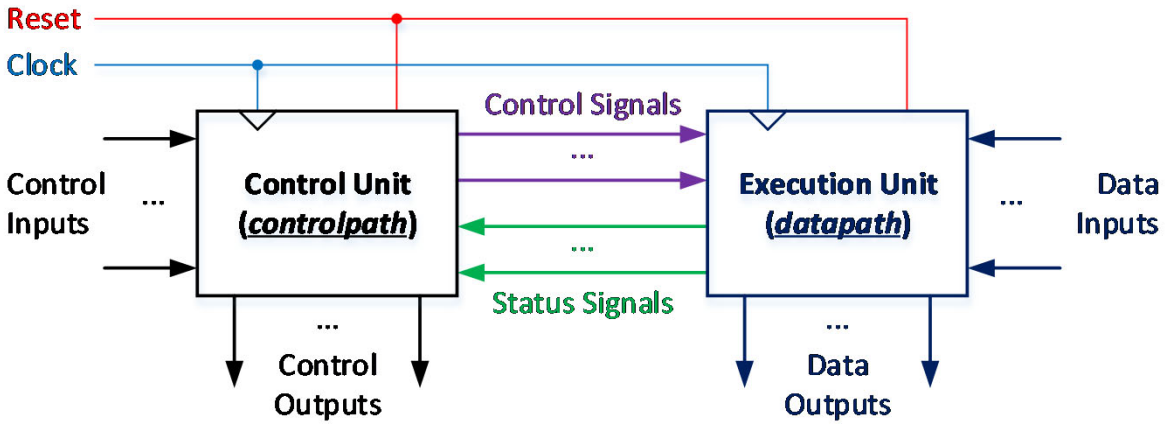
How many resources are required to build a 64 bit unsigned iterative multiplier?

Iterative/Combinatorial/Parallel vs. Sequential Multiplier

- Iterative/Combinatorial/Parallel implementation
  - Computes the result in a combinatorial way
- Sequential implementation
  - Determines the result in several clock cycles
  - Considers a sub-set of operand bits in each clock cycle
    - 1 bit of the multiplier x 1 one bit of the multiplicand
    - 1 bit of the multiplier x the entire multiplicand
- Iterative/Combinatorial/Parallel *vs.* Sequential implementations are in general a compromise between
  - Performance
  - Operating frequency
  - Amount of implementation resources
  - Power consumption
  - Cost

Computational System

- Datapath (execution unit)
  - Components
    - Functional
    - Routing
    - Storage
- Controlpath
  - Control unit
    - FSM(s)



- *Controlpath - datapath interconnection*
  - Control signals (controlpath → datapath)
  - Status signals (controlpath ← datapath)
- Datapath – Controlpath partition is very useful to design non trivial digital systems
- **Example: let's build a sequential multiplier based on architectural approach...**

Unsigned Multiplication

- Algorithm takes advantage of the distributive property in relation to the addition, allowing the multiplication to be decomposed into a succession of sums of partial products
- Consider the following product, where **M is the multiplicand** and **m is multiplier** with 4 bits:

$$R = M \cdot m$$

$$M \cdot m = M \cdot (m_3.2^3 + m_2.2^2 + m_1.2^1 + m_0.2^0)$$

$$M \cdot m = (M \cdot 2^3 \cdot m_3) + (M \cdot 2^2 \cdot m_2) + (M \cdot 2^1 \cdot m_1) + (M \cdot 2^0 \cdot m_0)$$

$$M \cdot m = ((M \cdot 2^3) \cdot m_3) + ((M \cdot 2^2) \cdot m_2) + ((M \cdot 2^1) \cdot m_1) + ((M.2^0) \cdot m_0)$$

Unsigned Multiplication

$$M \cdot m = ((M \cdot 2^3) \cdot m_3) + ((M \cdot 2^2) \cdot m_2) + ((M \cdot 2^1) \cdot m_1) + ((M.2^0) \cdot m_0)$$

- Multiplying by two (or by a power of two) corresponds to shifting the number multiplied to the left (**shift left**) as many bits as the power of two involved
- On the other hand, if m_n is equal to "0", the corresponding partial product will also be zero, and if it is "1", the same partial product will be equal to the multiplying left to the left of n bits

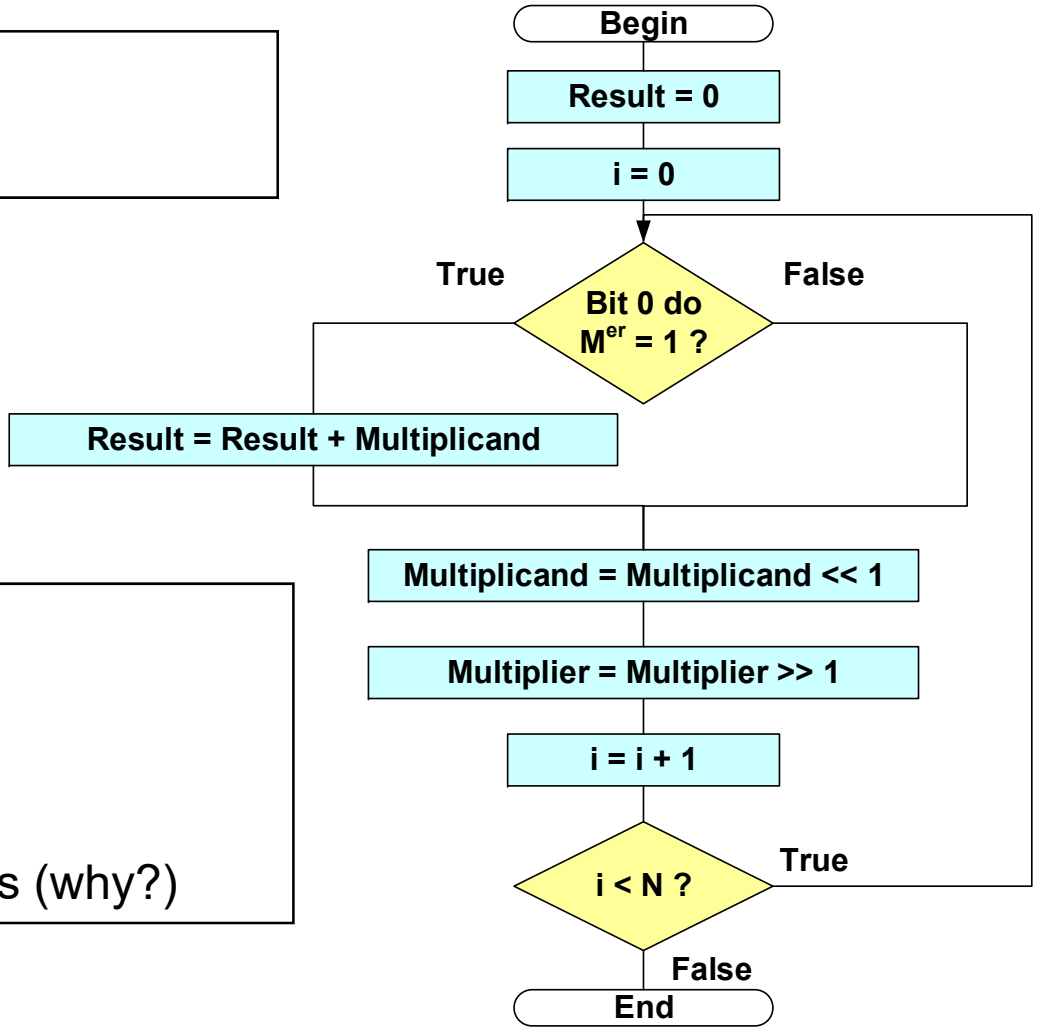
```
   1100
  x1101
  ----
   1100
  00000
 110000
+1100000
--------
 10011100
```

For each bit at "1" of the multiplier, the multiplicand is added to the left shift result of a number of bits equal to the position of "1" in the multiplier

Sequential Algorithm of the Unsigned Multiplication of **N** bits Operands

Operands: **N** bits

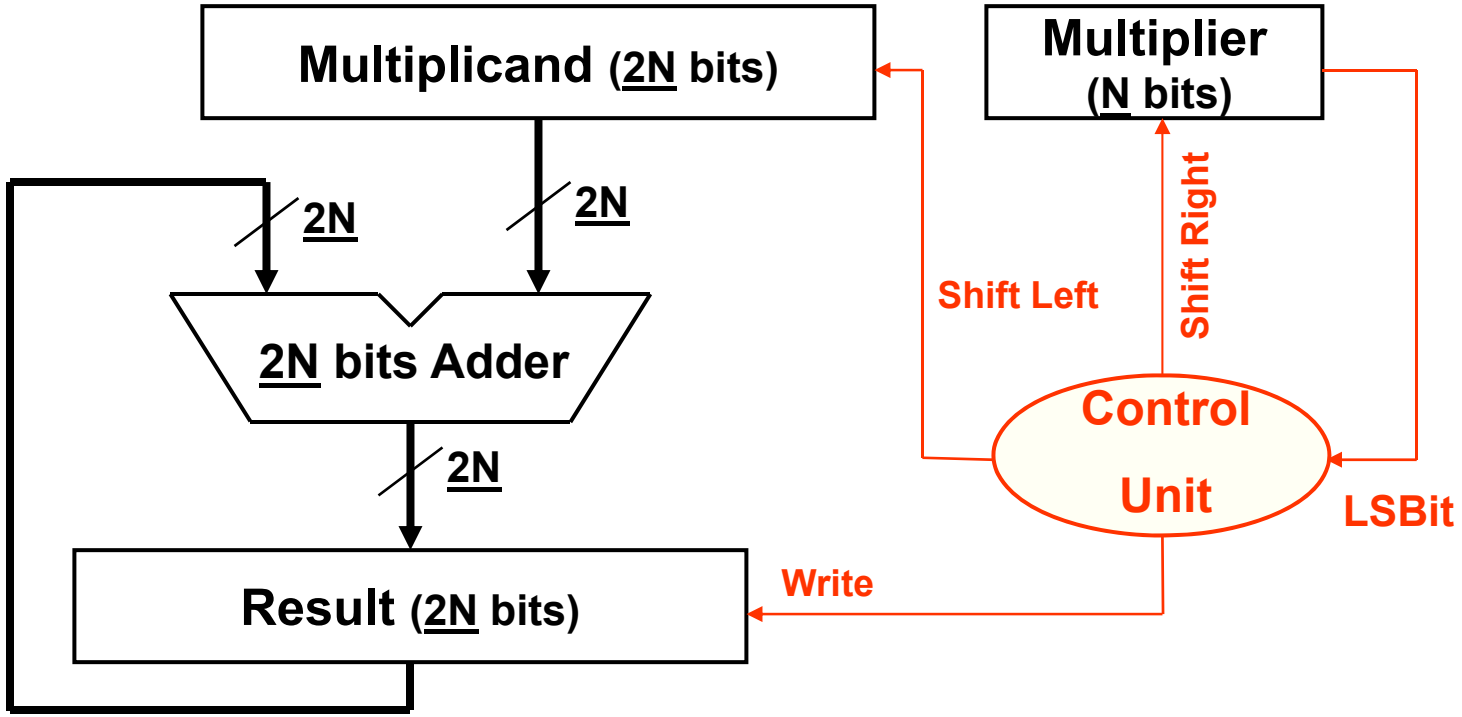
Result: **2N** bits



Required registers:

- **Result**: **2N** bits
- **Multiplier:** **N** bits
- **Multiplicand**: **2N** bits (why?)

Architecture of a N bits Operands Sequential Unsigned Multiplier



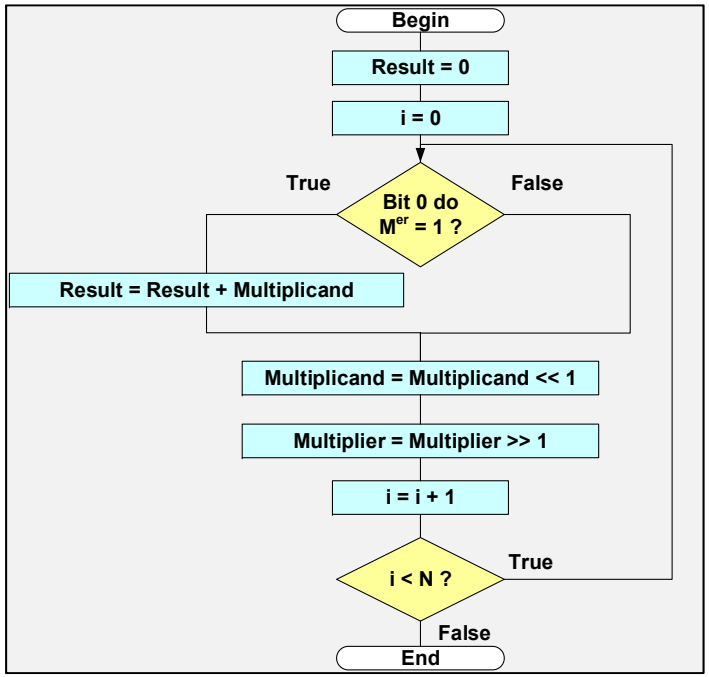
- The Adder and the Multiplication and Result registers operate with 2N bits
- The clock signal is not represented (it is implied) and synchronizes
  - writes in the multiplying, multiplier and result registers
  - state transitions of the control unit

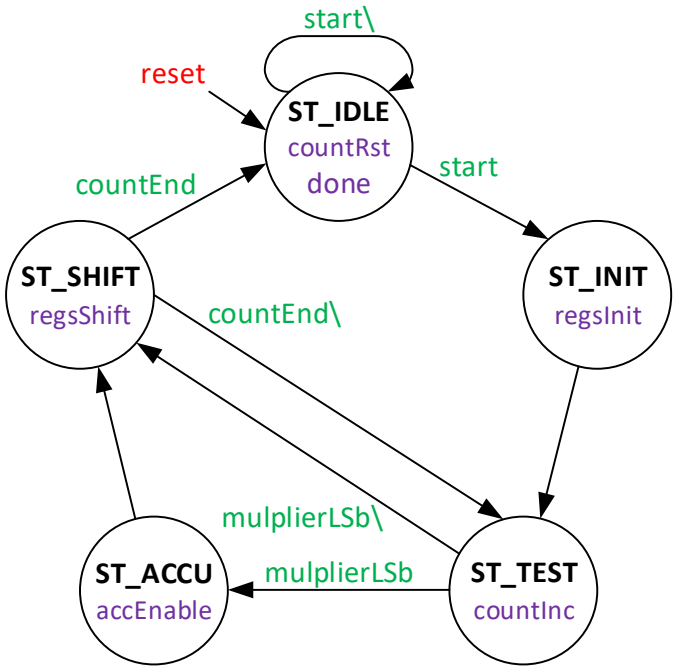
Example with 4 bits of the Sequential Unsigned Multiplication

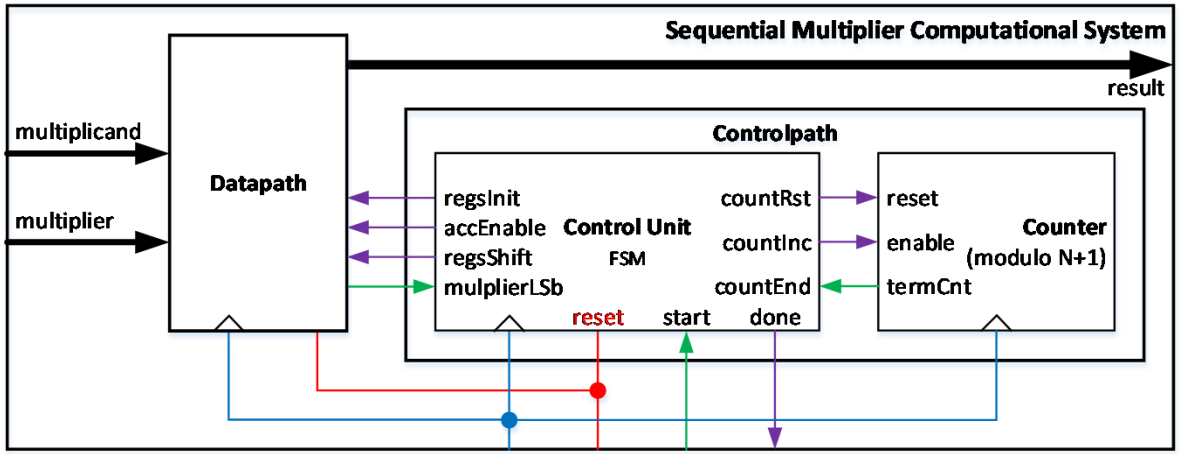
- With 4-bit operands, the result will have a maximum size of 8 bits
- For the implementation of a 4-bit multiplier, which applies the algorithm of the previous slide, the necessary registers are:
  - **result**: 8-bit register
  - **multipland**: 8 bit register (initially the most significant bits are set to 0000)
  - **multiplier**: 4-bit register

					1	1	0	0	
				x	1	1	0	1	
	0	0	0	0	0	0	0	0	Initial result
+	0	0	0	0	1	1	0	0	1.mdo.2^0
	0	0	0	0	1	1	0	0	
+	0	0	0	0	0	0	0	0	0.mdo.2^1
	0	0	0	0	1	1	0	0	
+	0	0	1	1	0	0	0	0	1.mdo.2^2
	0	0	1	1	1	1	0	0	
+	0	1	1	0	0	0	0	0	1.mdo.2^3
	1	0	0	1	1	1	0	0	
	1	0	0	1	1	1	0	0	Final result

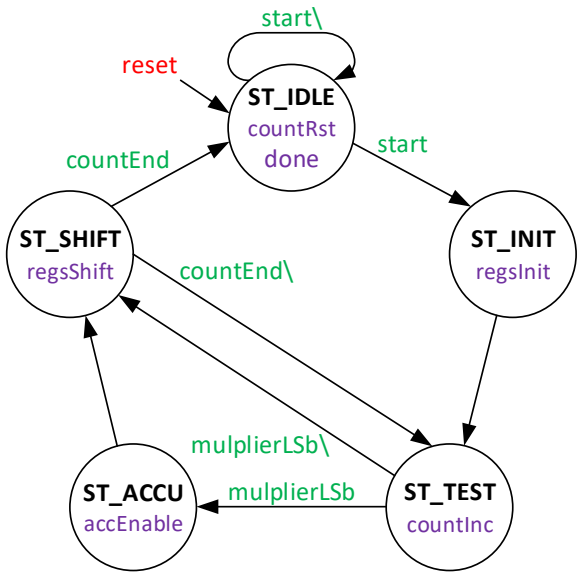
Sequential Multiplier Control Unit FSM



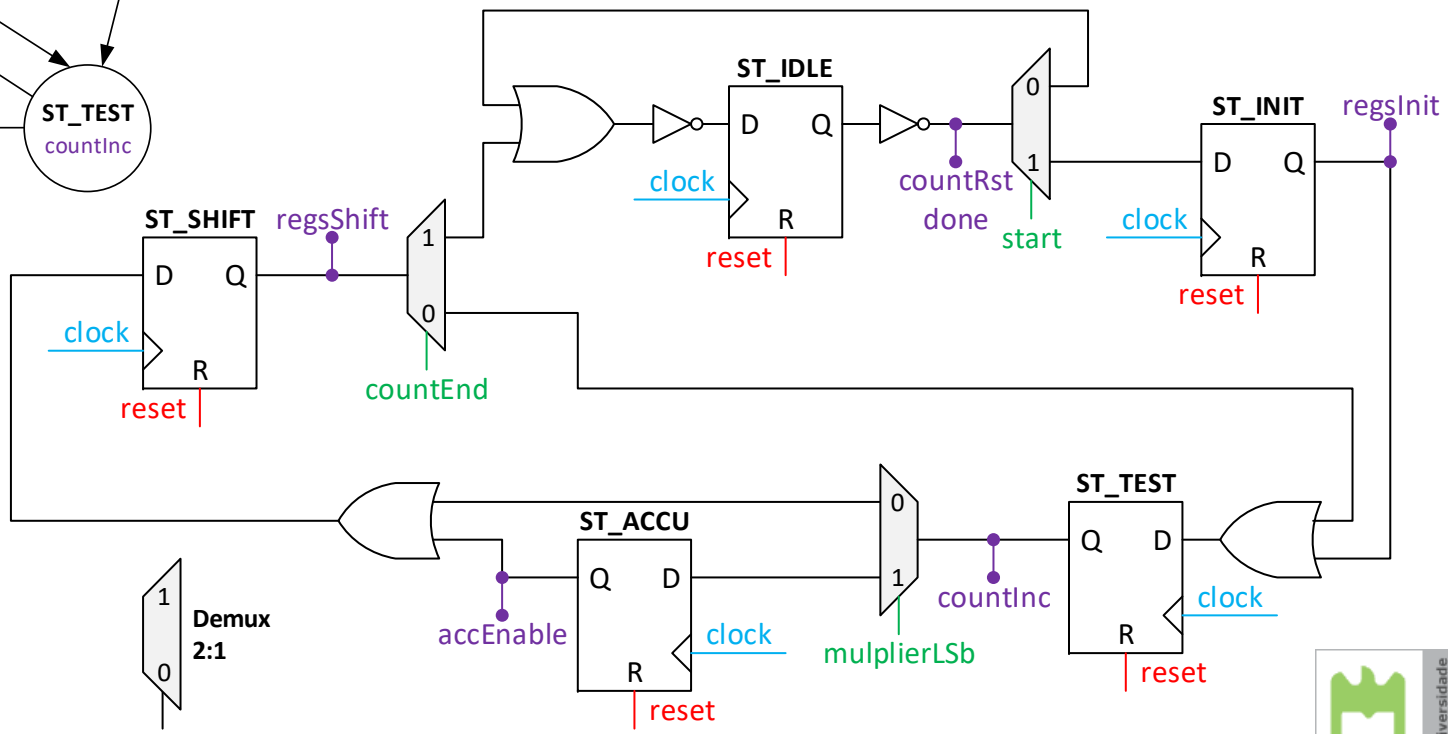




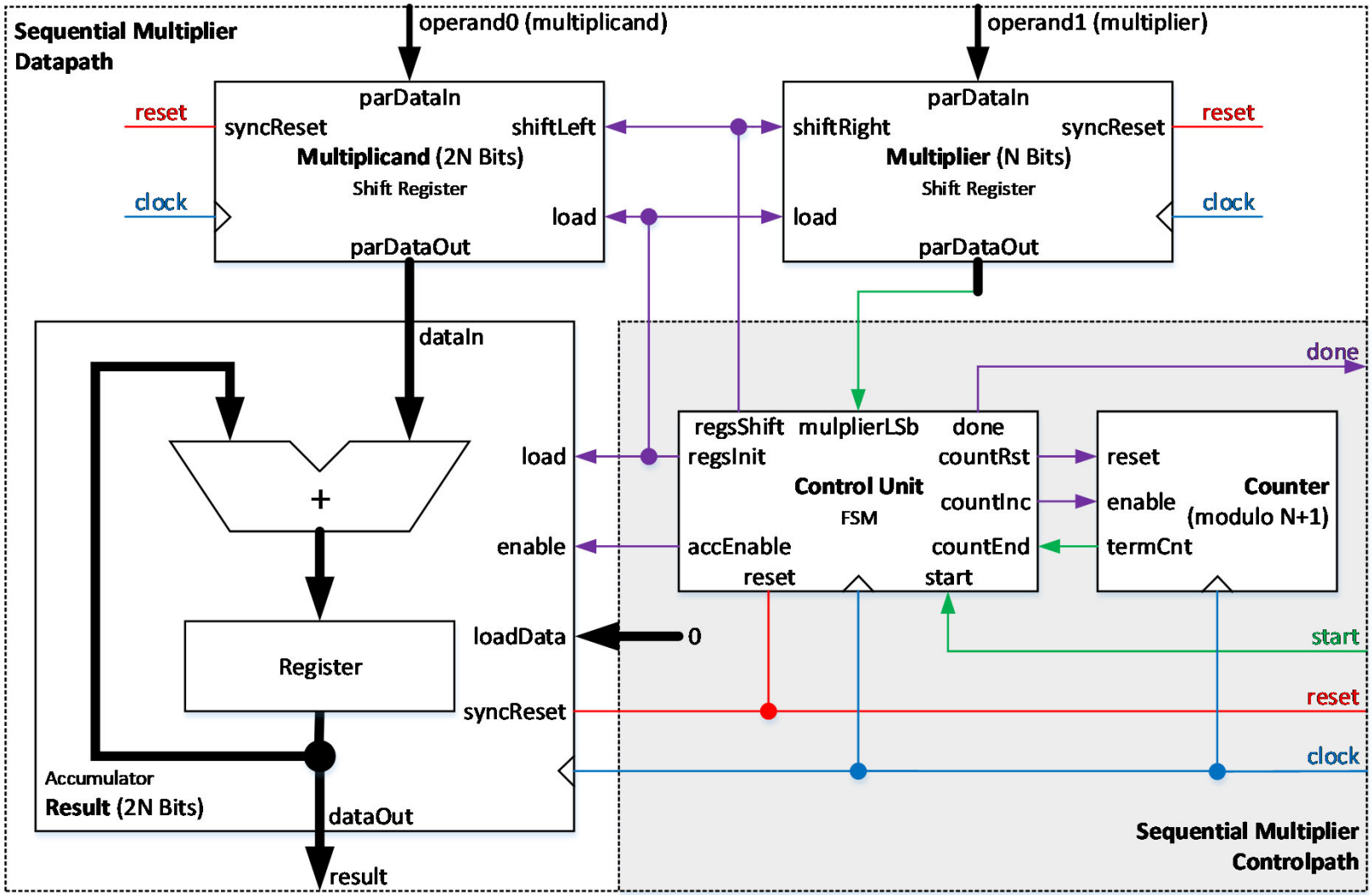
Sequential Multiplier Control Unit



Synthesis using one-hot state encoding



Sequential Multiplier Detailed Controlpath+Datapath



Conclusion

- At the end of this lecture and corresponding lab, it is fundamental to:
  - Distinguish iterative from sequential circuits
  - Analyse and design small to medium complexity digital systems composed of a control unit and a datapath

Reading chapter 8 (4th ed.) or chapter 11 (5th ed.) of *John F. Wakerly, "Digital Design – Principles and Practices", Pearson – Prentice Hall*, is highly recommended.