

## ECE 545—Digital System Design with VHDL

### Sorting ASM Example

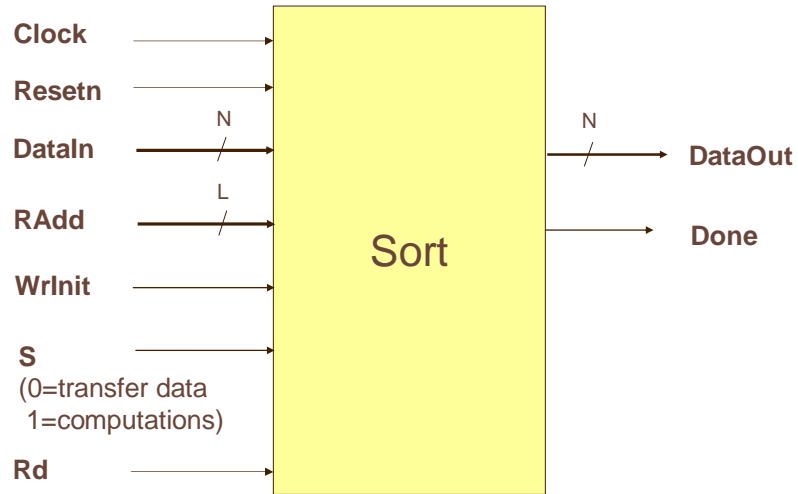
1

### Specification

- Given  $k$  unsigned  $N$ -bit numbers stored in a set of registers  $R_0$  to  $R_{k-1}$ , design a circuit that can sort the list in ascending order. After the sort, the smallest value should be in  $R_0$  and the largest value should be in  $R_{k-1}$ .
- In this example,  $k = 4$  numbers

2

## Sorting – Required Interface



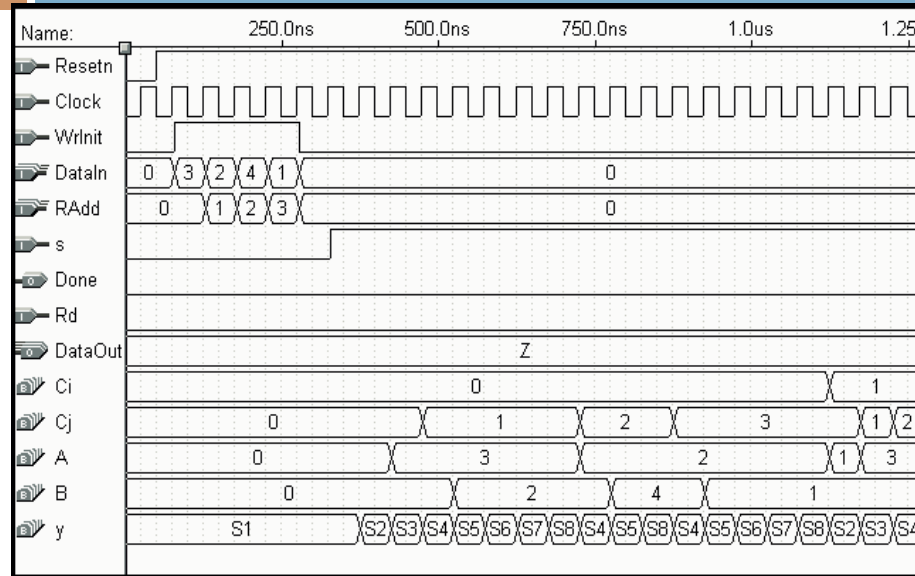
3

## Explanation of ports

- Inputs
  - **Clock**: the circuit clock
  - **Resetn**: active-low asynchronous circuit reset
  - **DataIn**: N-bit input. Data will be input into the system one value at a time using DataIn and RAdd.
  - **RAdd**: L-bit input, where L is  $\log_2(k)$ . k is the number of inputs to sort. RAdd stands for Register Address. It will be use to read in the data and read out the data.
  - **WrInit**: Set high to write data into the system initially.
  - **S**: system mode.
    - 0 = write data initially or read results.
    - 1 = do computations
  - **Rd**: Set high to read final data results.
- Outputs
  - **DataOut**: N-bit output. Sorted data will be output from the system one value at a time using DataOut and RAdd.
  - **Done**: Indicates the sort operation is done and sorted data is ready to be read out on DataOut

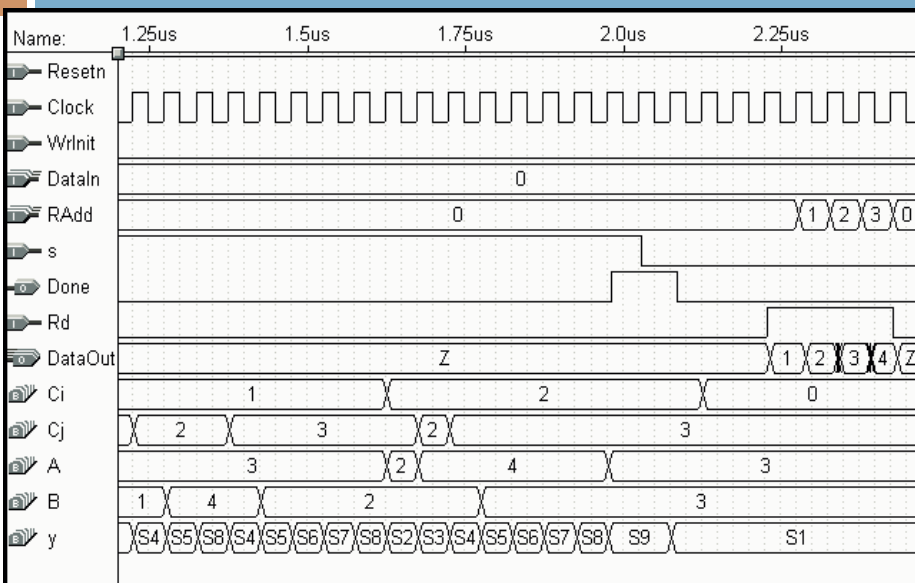
4

## Simulation results for the sort operation (1) Loading the registers and starting sorting



5

## Simulation results for the sort operation (2) Completing sorting and reading out registers



6

## Sorting - Example

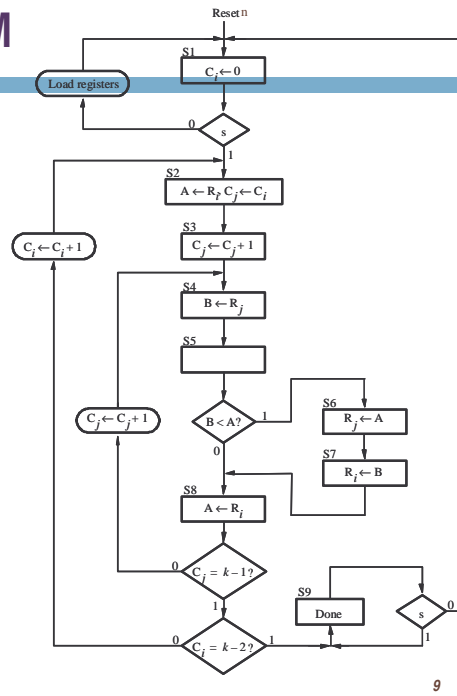
		During Sorting						After sorting
Before sorting		i=0	i=0	i=0	i=1	i=1	i=2	
		j=1	j=2	j=3	j=2	j=3	j=3	
Address								
0	3	<span style="border: 1px solid black; padding: 2px;">3</span>	<span style="border: 1px solid black; padding: 2px;">2</span>	<span style="border: 1px solid black; padding: 2px;">2</span>	1	1	1	1
1	2	<span style="border: 1px solid black; border-radius: 50%; padding: 2px;">2</span>	3	3	<span style="border: 1px solid black; padding: 2px;">3</span>	<span style="border: 1px solid black; padding: 2px;">3</span>	2	2
2	4	4	<span style="border: 1px solid black; border-radius: 50%; padding: 2px;">4</span>	4	<span style="border: 1px solid black; border-radius: 50%; padding: 2px;">4</span>	4	<span style="border: 1px solid black; padding: 2px;">4</span>	3
3	1	1	1	<span style="border: 1px solid black; border-radius: 50%; padding: 2px;">1</span>	2	<span style="border: 1px solid black; border-radius: 50%; padding: 2px;">2</span>	<span style="border: 1px solid black; border-radius: 50%; padding: 2px;">3</span>	4

Legend:    position of memory indexed by i     $R_i$     position of memory indexed by j     $R_j$

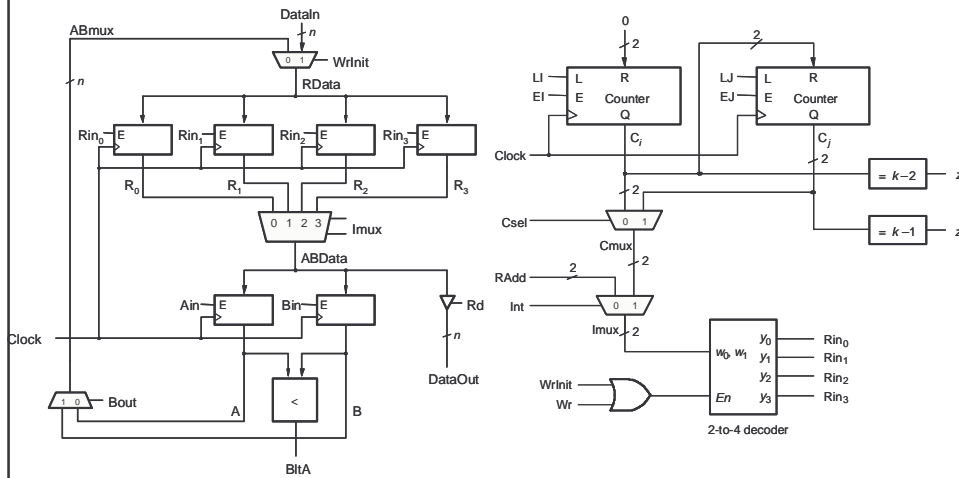
## Step 1: Pseudocode for the sort operation

<pre> [load input data] for i = 0 to k - 2 do   A = R<sub>i</sub> ;   for j = i + 1 to k - 1 do     B = R<sub>j</sub> ;     if B &lt; A then       R<sub>i</sub> = B ;       R<sub>j</sub> = A ;       A = R<sub>i</sub> ;     end if ;   end for ; end for ; [read output data]         </pre>	<pre> <b>FOR k = 4</b> [load input data] for i = 0 to 2 do   A = R<sub>i</sub> ;   for j = i + 1 to 3 do     B = R<sub>j</sub> ;     if B &lt; A then       R<sub>i</sub> = B ;       R<sub>j</sub> = A ;       A = R<sub>i</sub> ;     end if ;   end for ; end for ; [read output data]         </pre>
---	--

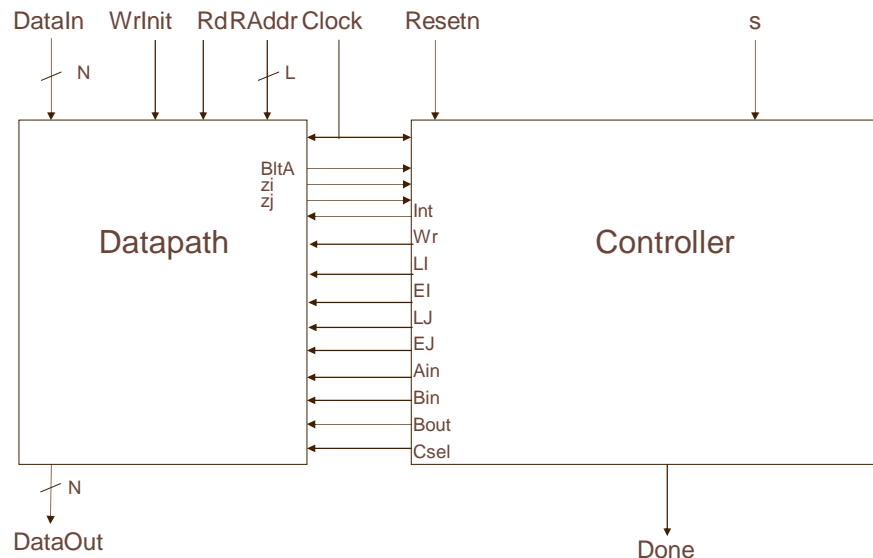
## Step 2: High-Level ASM



## Step 3: Datapath Block Diagram



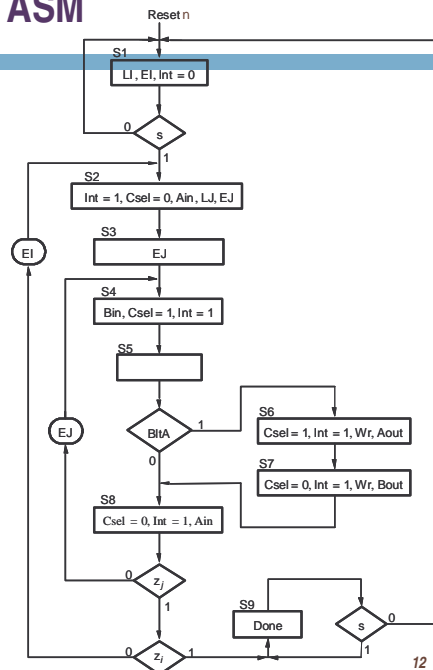
## Step 4: Top-Level Interface Diagram



11

## Step 5: Detailed Controller ASM

- Inputs to the controller are in diamonds (decision box) or external
- Outputs from controller are in squares (state box) or ovals (conditional output box)
- All transactions in a state box or conditional output box must complete in one clock cycle



12