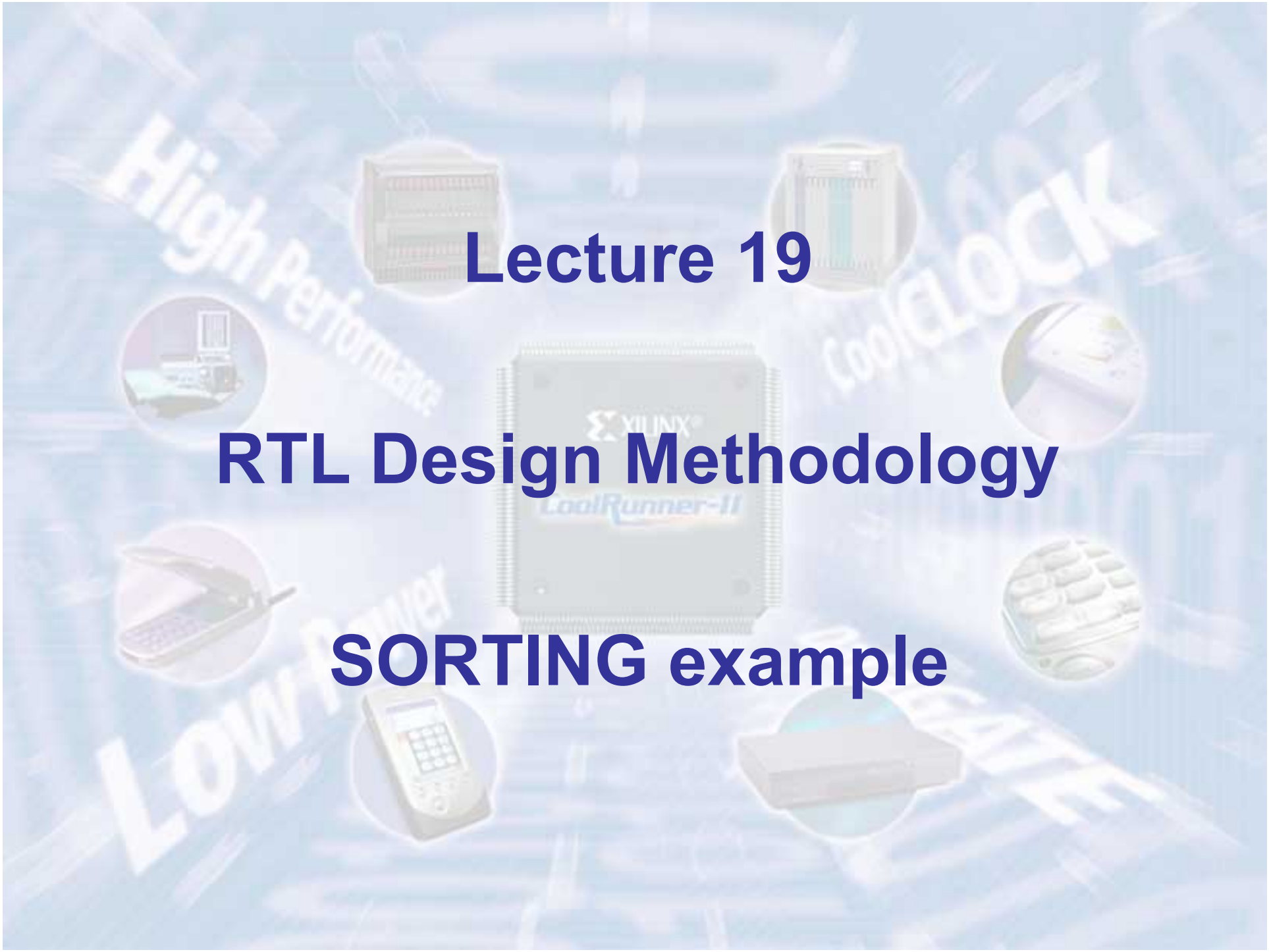
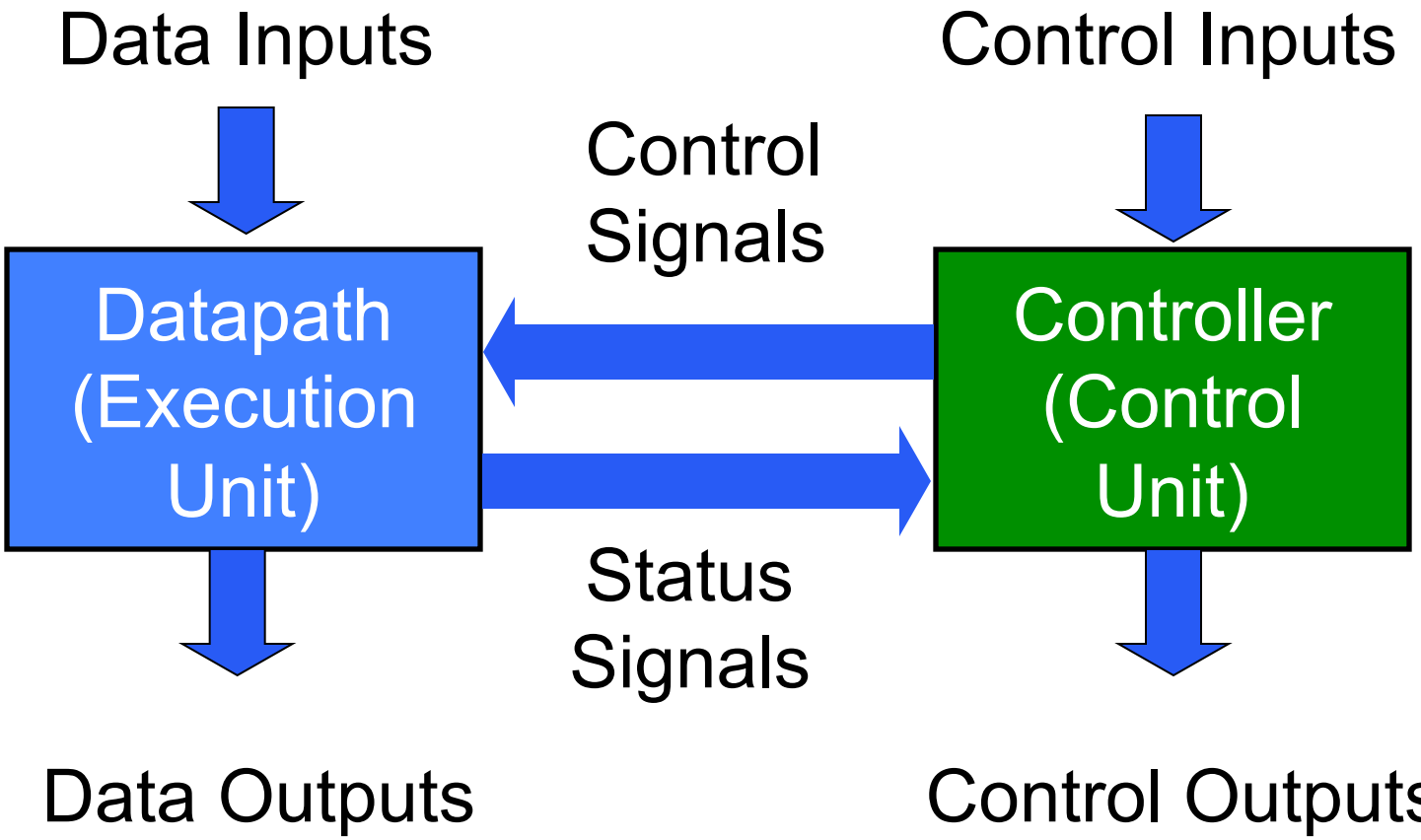


Lecture 19

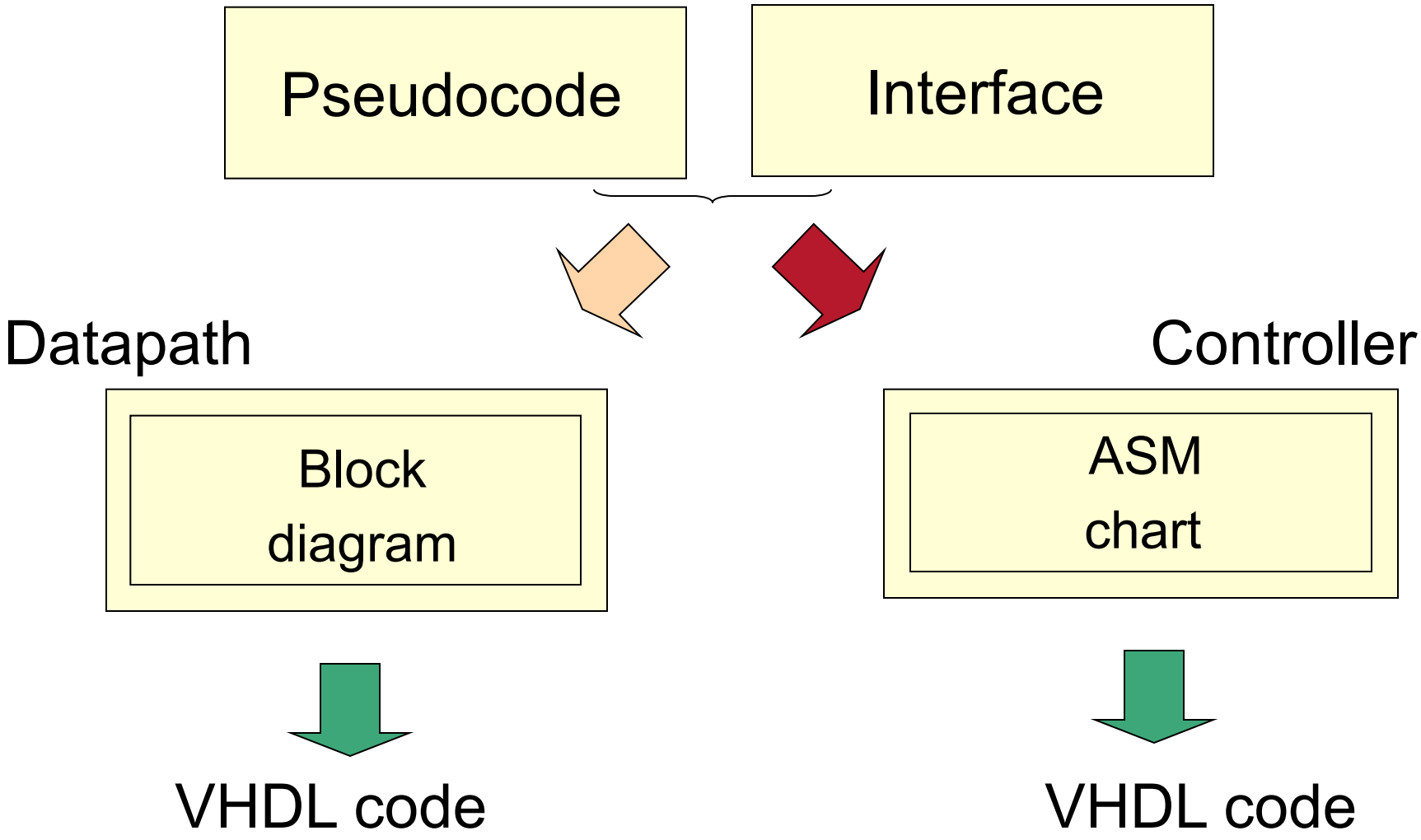
RTL Design Methodology

SORTING example

Structure of a Typical Digital System



Hardware Design with RTL VHDL

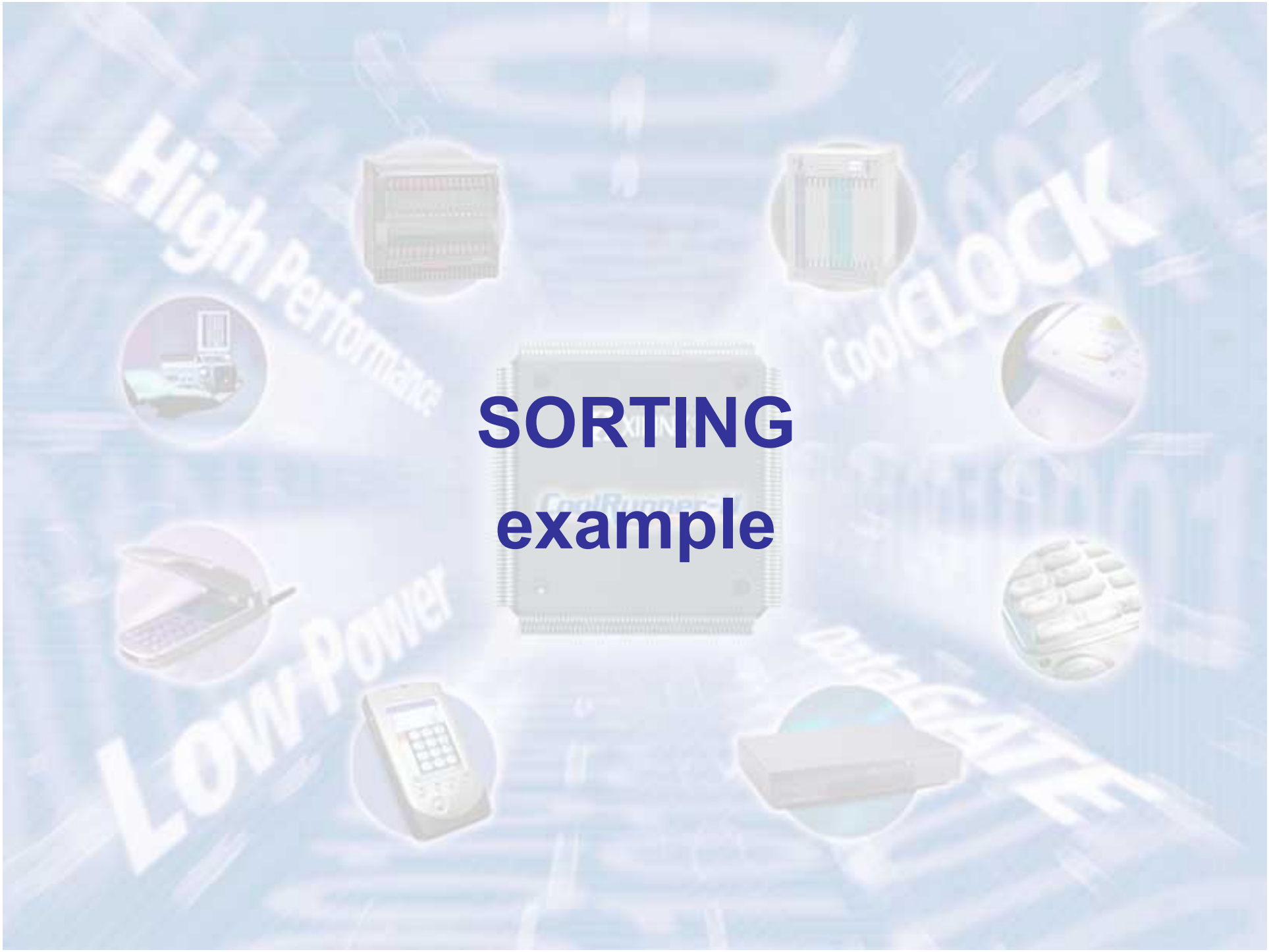


Steps of the Design Process

1. Text description
2. Interface
3. Pseudocode
4. Block diagram of the Datapath
5. Interface divided into Datapath and Controller
6. ASM chart of the Controller
7. RTL VHDL code of the Datapath, Controller, and Top-Level Unit
8. Testbench for the Datapath, Controller, and Top-Level Unit
9. Functional simulation and debugging
10. Synthesis and post-synthesis simulation
11. Implementation and timing simulation
12. Experimental testing using FPGA board

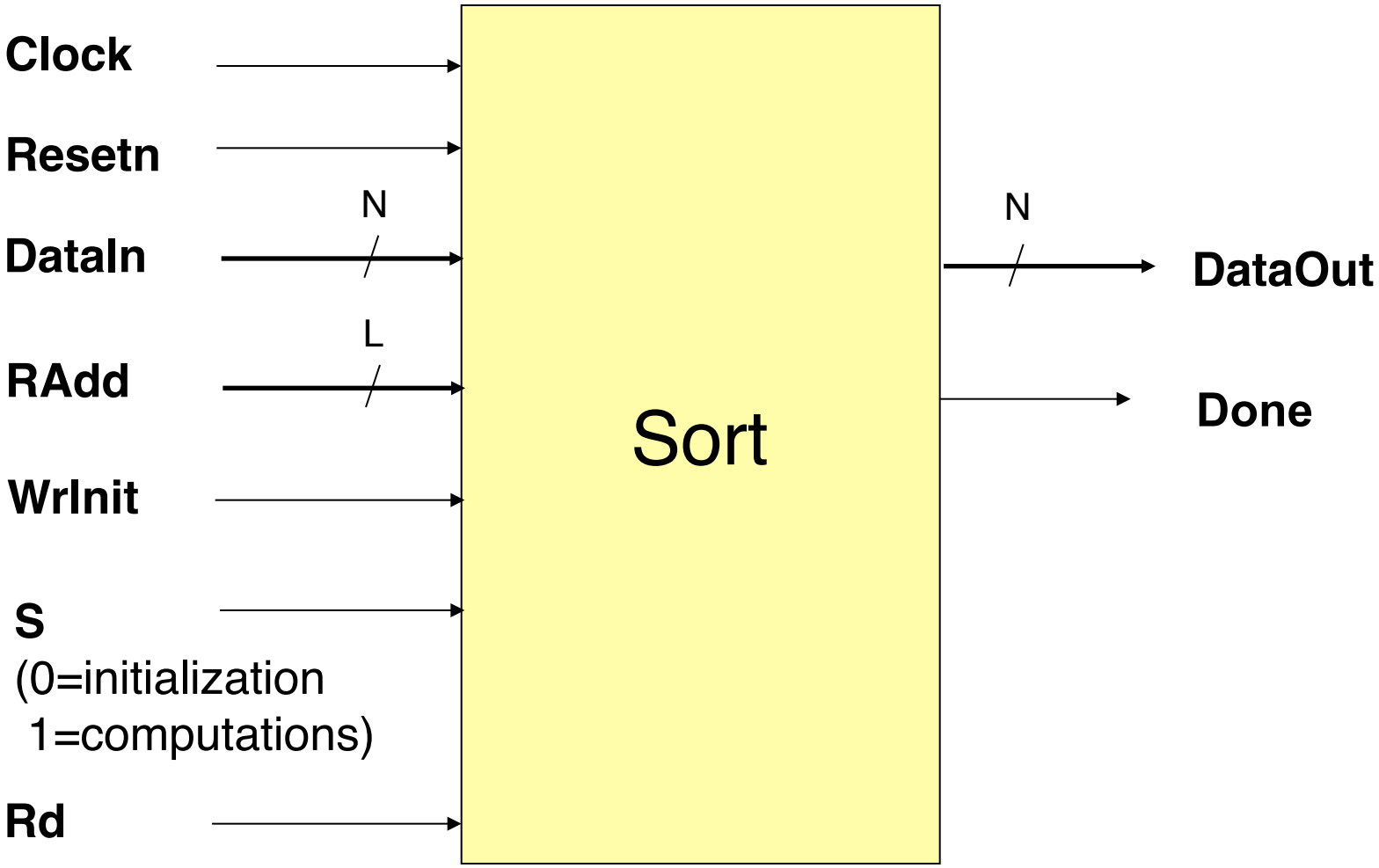
Steps of the Design Process Introduced in Class Today

1. Text description
2. Interface
3. Pseudocode
4. **Block diagram of the Datapath**
5. **Interface divided into Datapath and Controller**
6. **ASM chart of the Controller**
7. RTL VHDL code of the Datapath, Controller, and Top-level Unit
8. Testbench for the Datapath, Controller, and Top-Level Unit
9. Functional simulation and debugging
10. Synthesis and post-synthesis simulation
11. Implementation and timing simulation
12. Experimental testing using FPGA board



SORTING
example

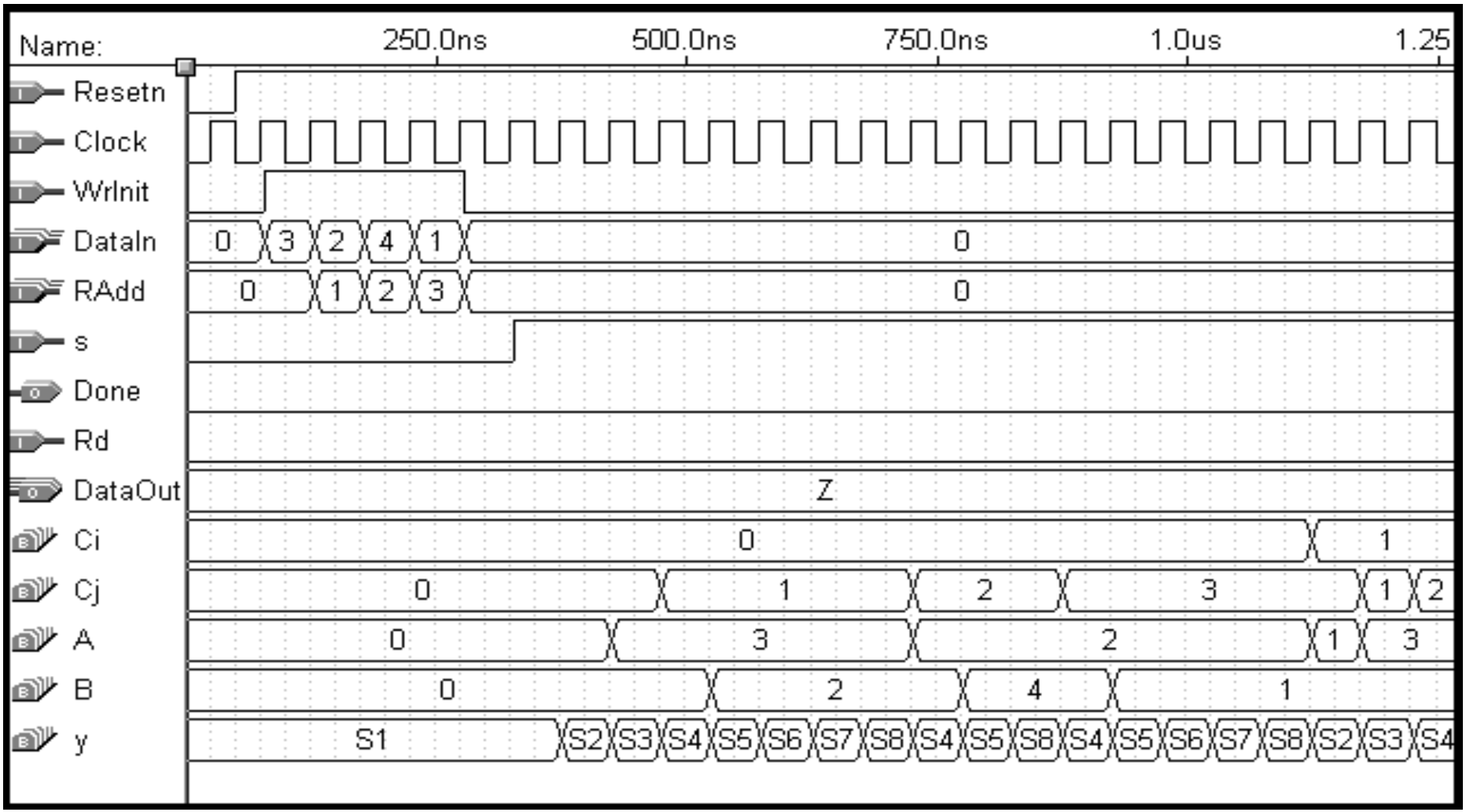
Sorting - Required Interface



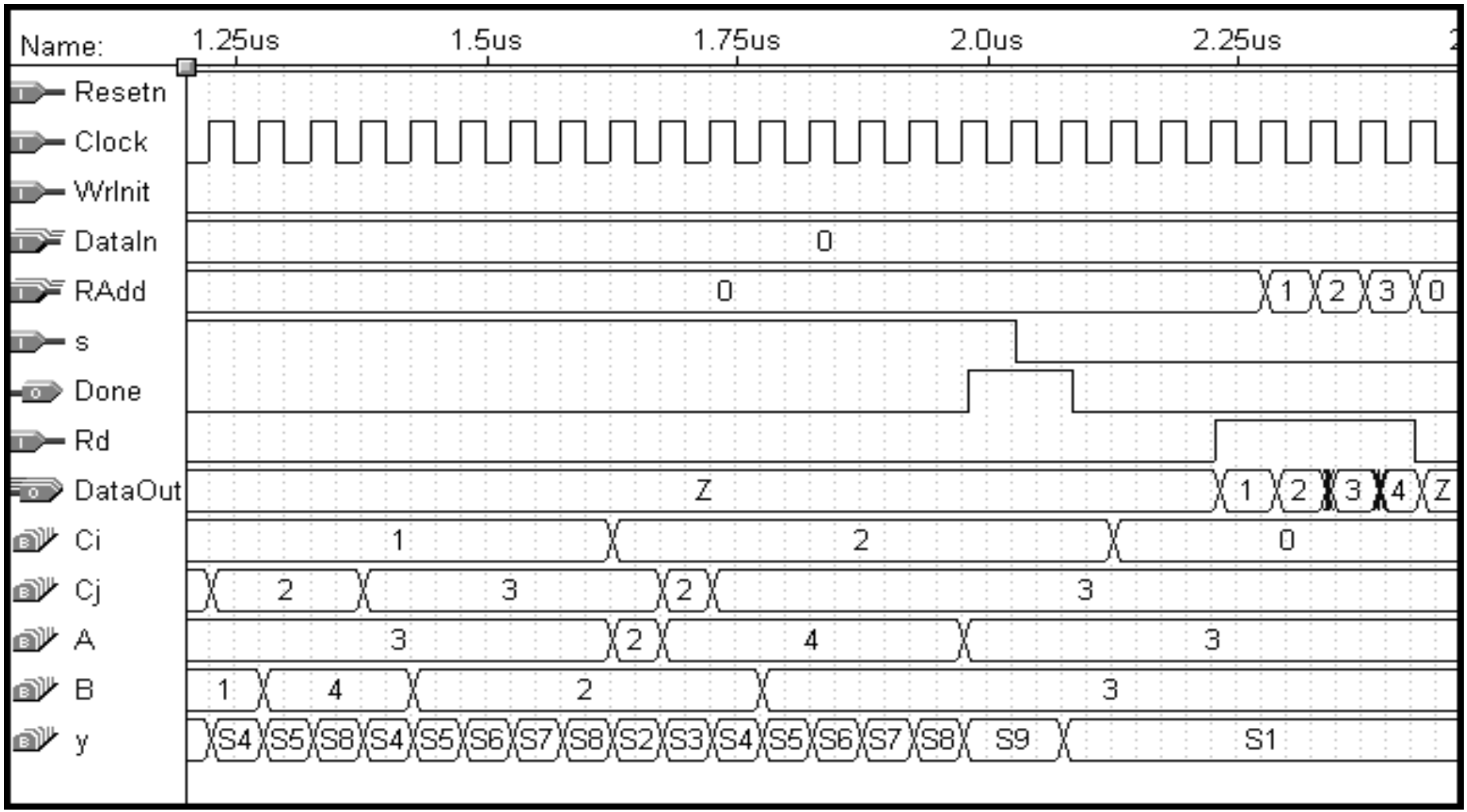
Sorting - Required Interface

Port	Width	Meaning
Clock	1	System clock
Resetn	1	System reset – clears internal registers. Active low.
DataIn	N	Input data bus
RAdd	L	Address of the internal memory where input data is stored
WrInit	1	Synchronous write control signal
s	1	Operating mode: 0 = initialization, 1 = computations.
Rd	1	Read enable. 0 = high impedance on the output bus, 1 = valid output on the output data bus.
DataOut	N	Output data bus used to read results
Done	1	Asserted when all results are ready

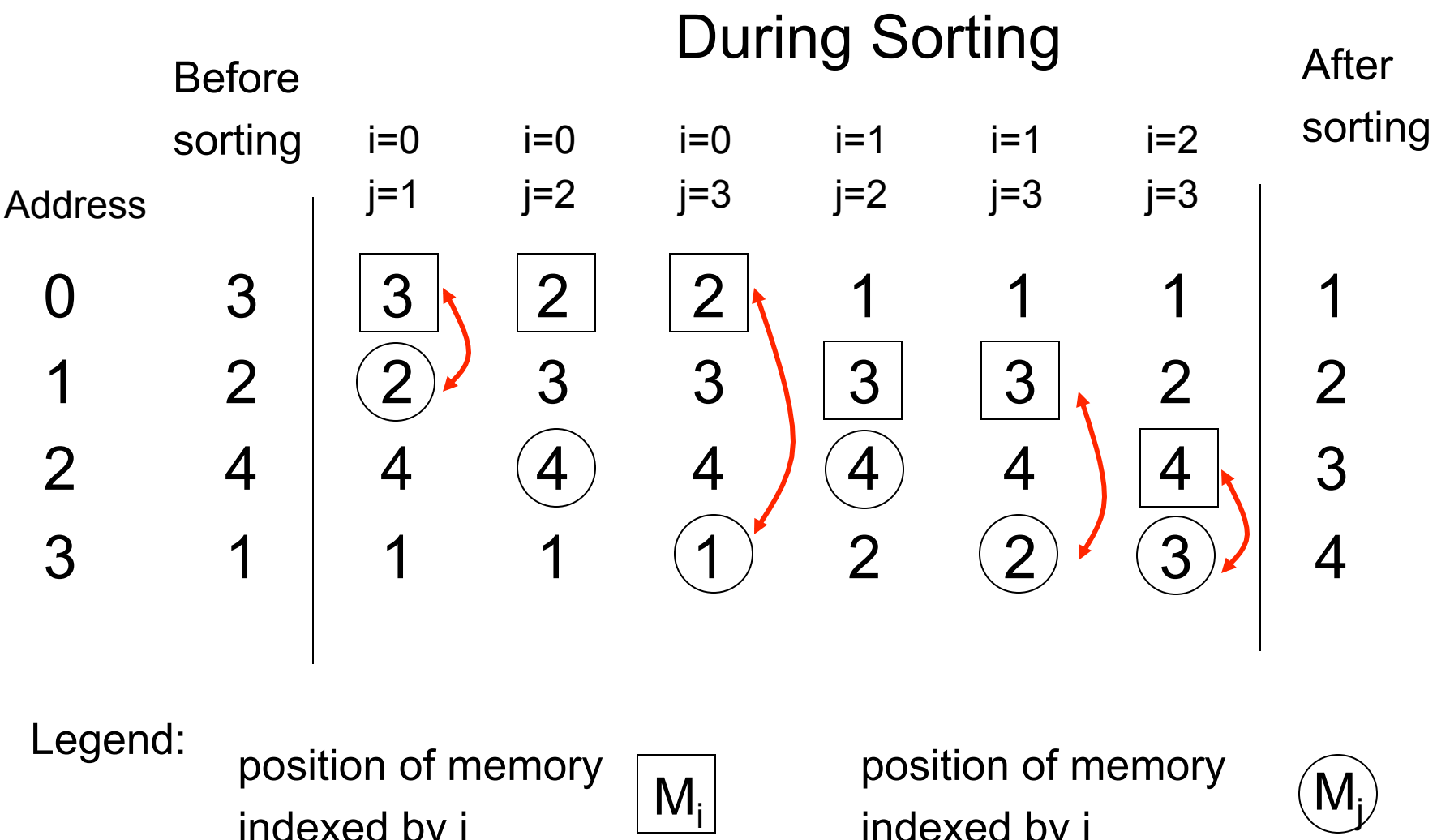
Simulation results for the sort operation (1)
Loading memory and starting sorting



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



Sorting - Example



Legend:

position of memory indexed by i M_i

position of memory indexed by j M_j

Pseudocode

FOR k = 4

```
[load input data]
wait for s=1
 for i = 0 to 2 do
       A = Mi ;
       for j = i + 1 to 3 do
             B = Mj ;
             if B < A then
                   Mi = B ;
                   Mj = A ;
                   A = Mi ;
             endif ;
       endfor;
 endfor;
Done
wait for s=0
[read output data]
go to the beginning
```

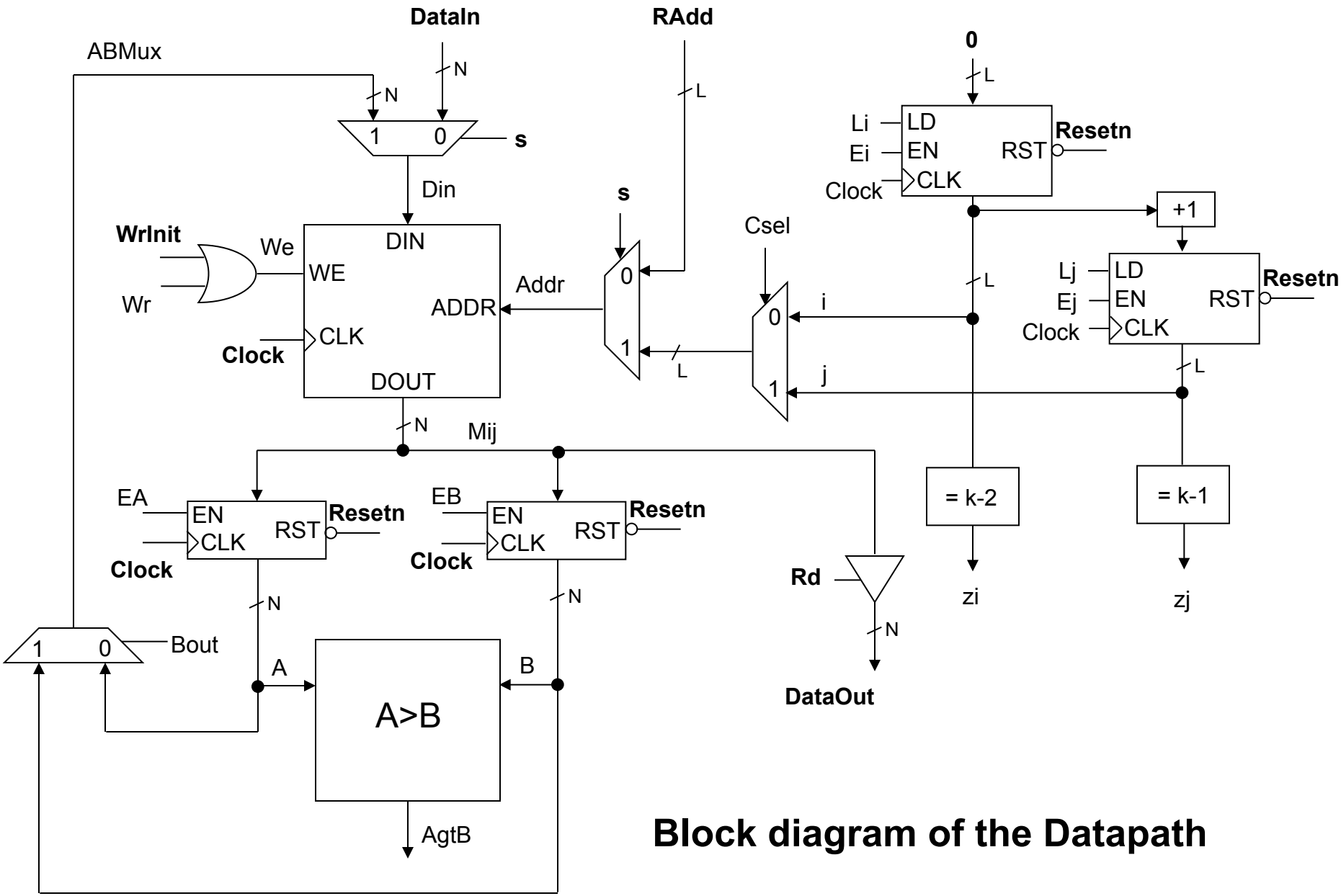
FOR any k ≥ 2

```
[load input data]
wait for s=1
 for i = 0 to k-2 do
       A = Mi ;
       for j = i + 1 to k – 1 do
             B = Mj ;
             if B < A then
                   Mi = B ;
                   Mj = A ;
                   A = Mi ;
             endif ;
       endfor;
 endfor;
 Done
 wait for s=0
 [read output data]
 go to the beginning
```

Pseudocode

```
wait for s=1
for i=0 to k-2 do
    A = M_i
    for j=i+1 to k-1 do
        B = M_j
        if A > B then
            M_i = B
            M_j = A
            A = M_i
        end if
    end for
end for
Done
wait for s=0
go to the beginning
```

SORTING
solutions



Block diagram of the Datapath

Interface with the division into Datapath and Controller