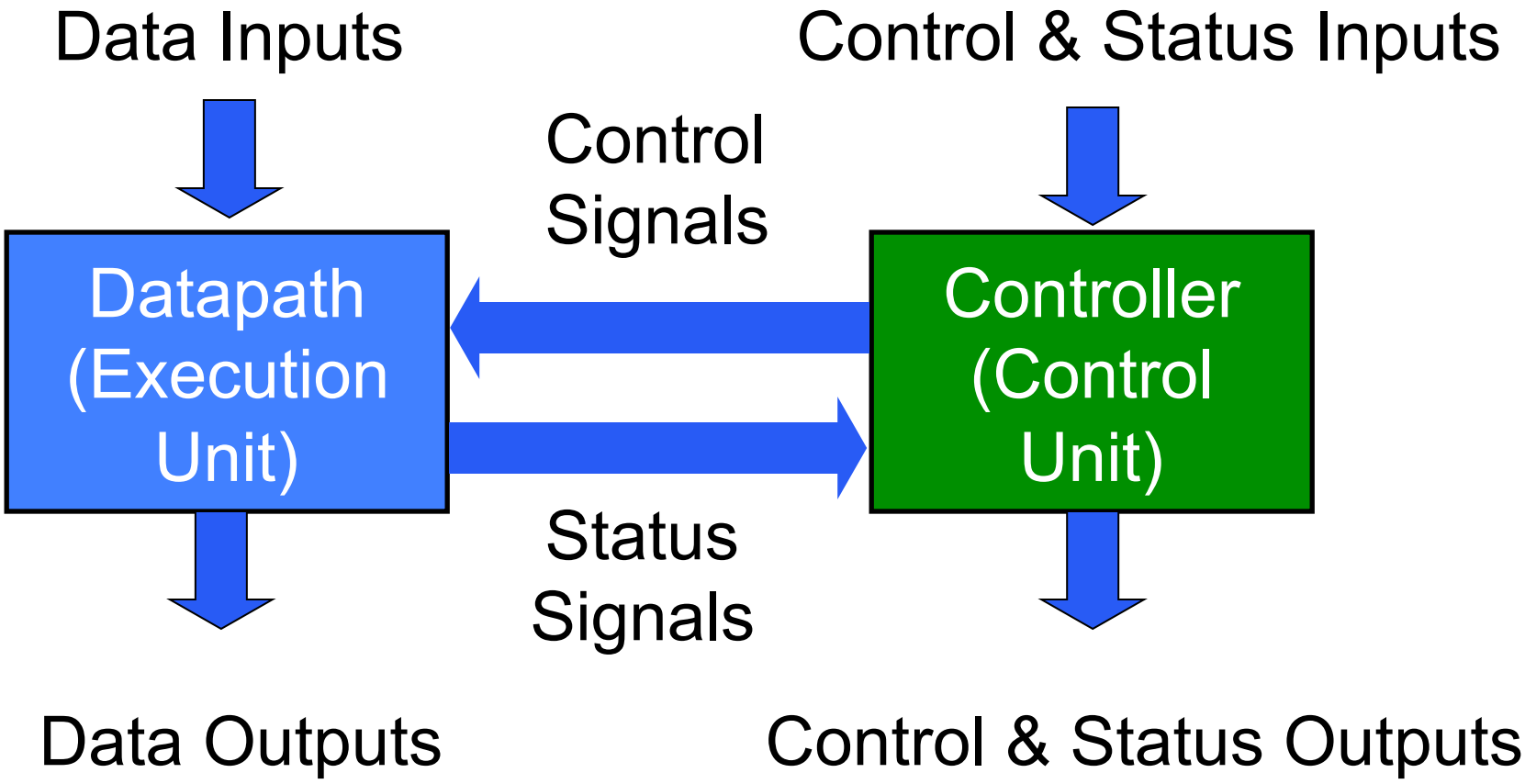
Lecture 2

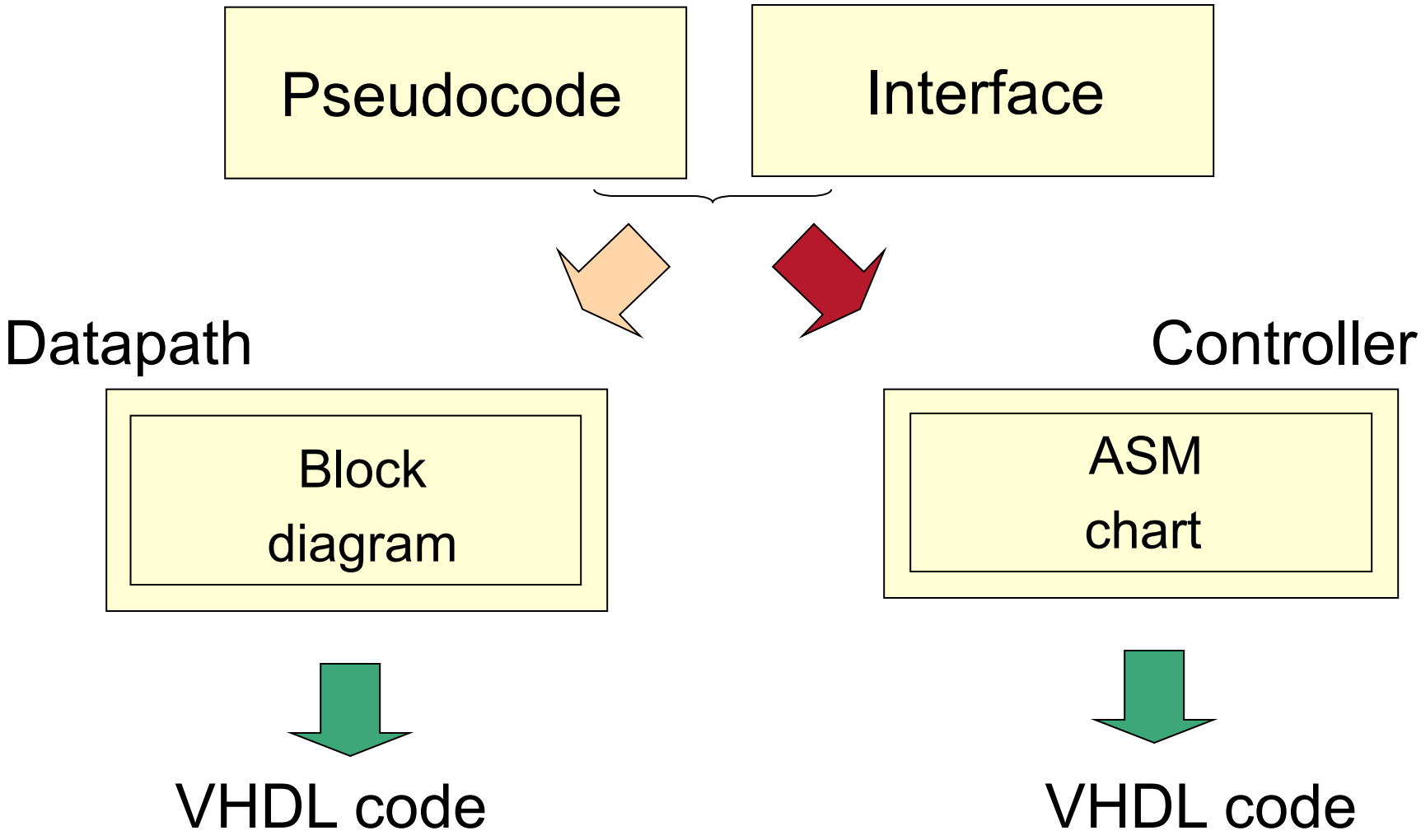
RTL Design Methodology

Transition from Pseudocode & Interface to a Corresponding Block Diagram

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

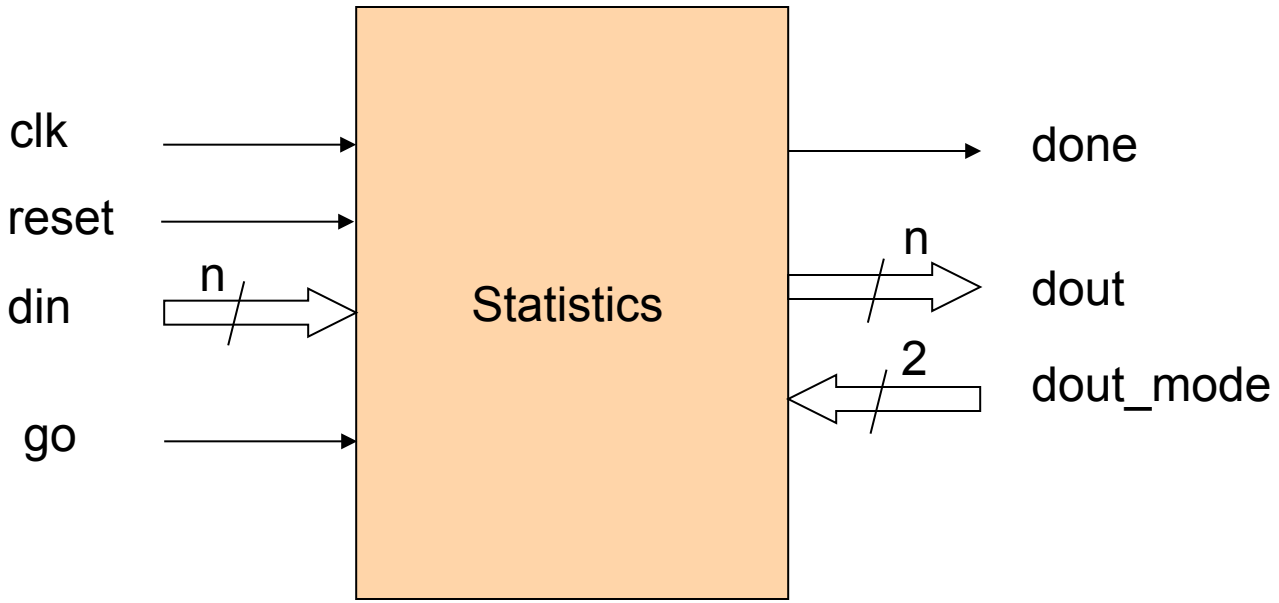


Class Exercise 1
STATISTICS

Pseudocode

```
no_1 = no_2 = no_3 = sum = 0
for i=0 to k-1 do
    sum = sum + din
    if  din > no_1 then
            no_3 = no_2
            no_2 = no_1
            no_1 = din
    elseif (din > no_2) then
            no_3 = no_2
            no_2 = din
    elseif (din > no_3) then
            no_3 = din
    end if
end for
avr = sum / k
```

Circuit Interface

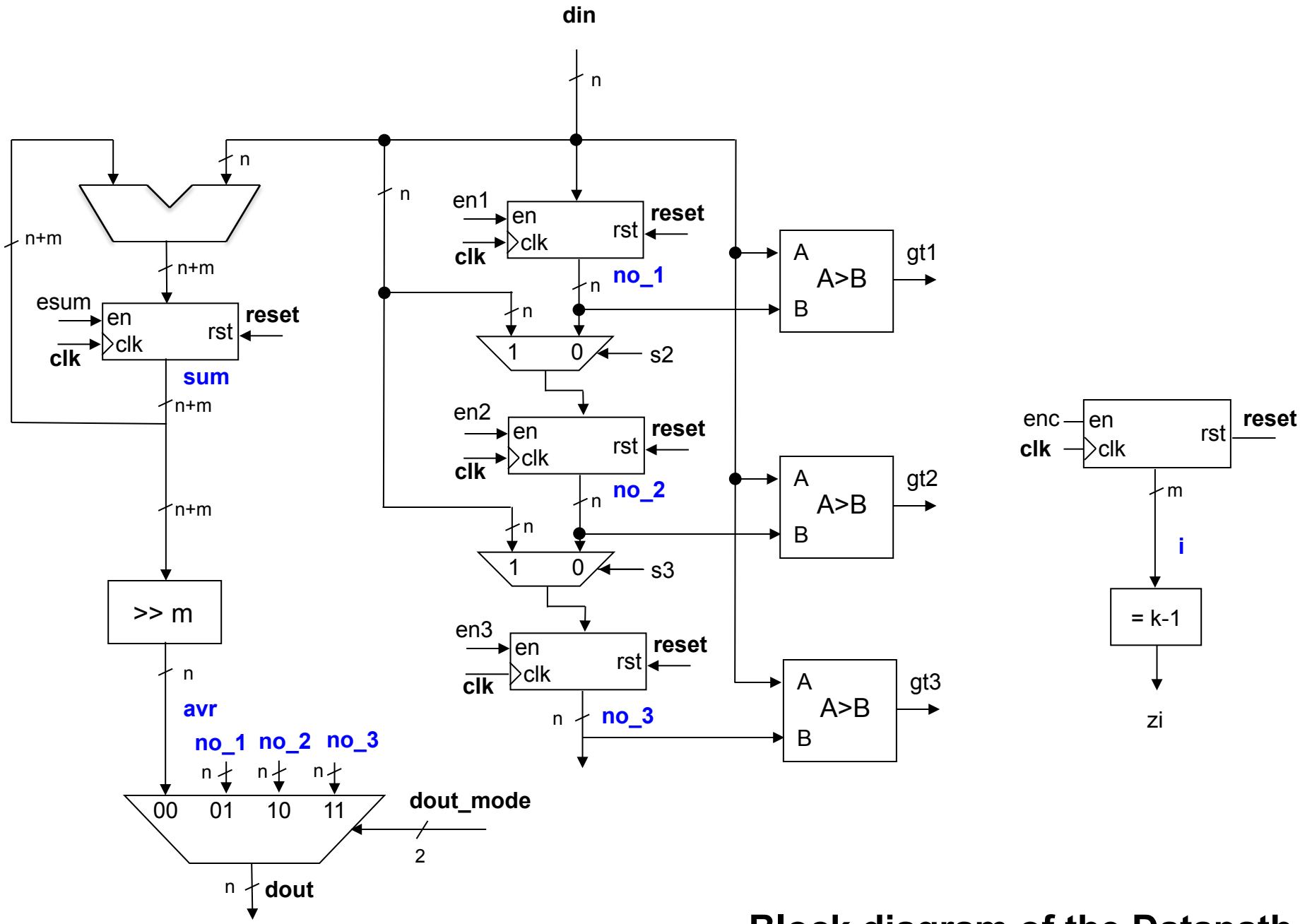


Interface Table

Port	Width	Meaning
clk	1	System clock.
reset	1	System reset.
din	n	Input Data.
go	1	Control signal indicating that the first input is ready. Active for one clock cycle.
done	1	Signal set to high after the output is ready.
dout	n	Output dependent on the dout_mode input.
dout_mode	2	Control signal determining value available at the output. 00: avr, 01: no_1, 10: no_2, 11: no_3.

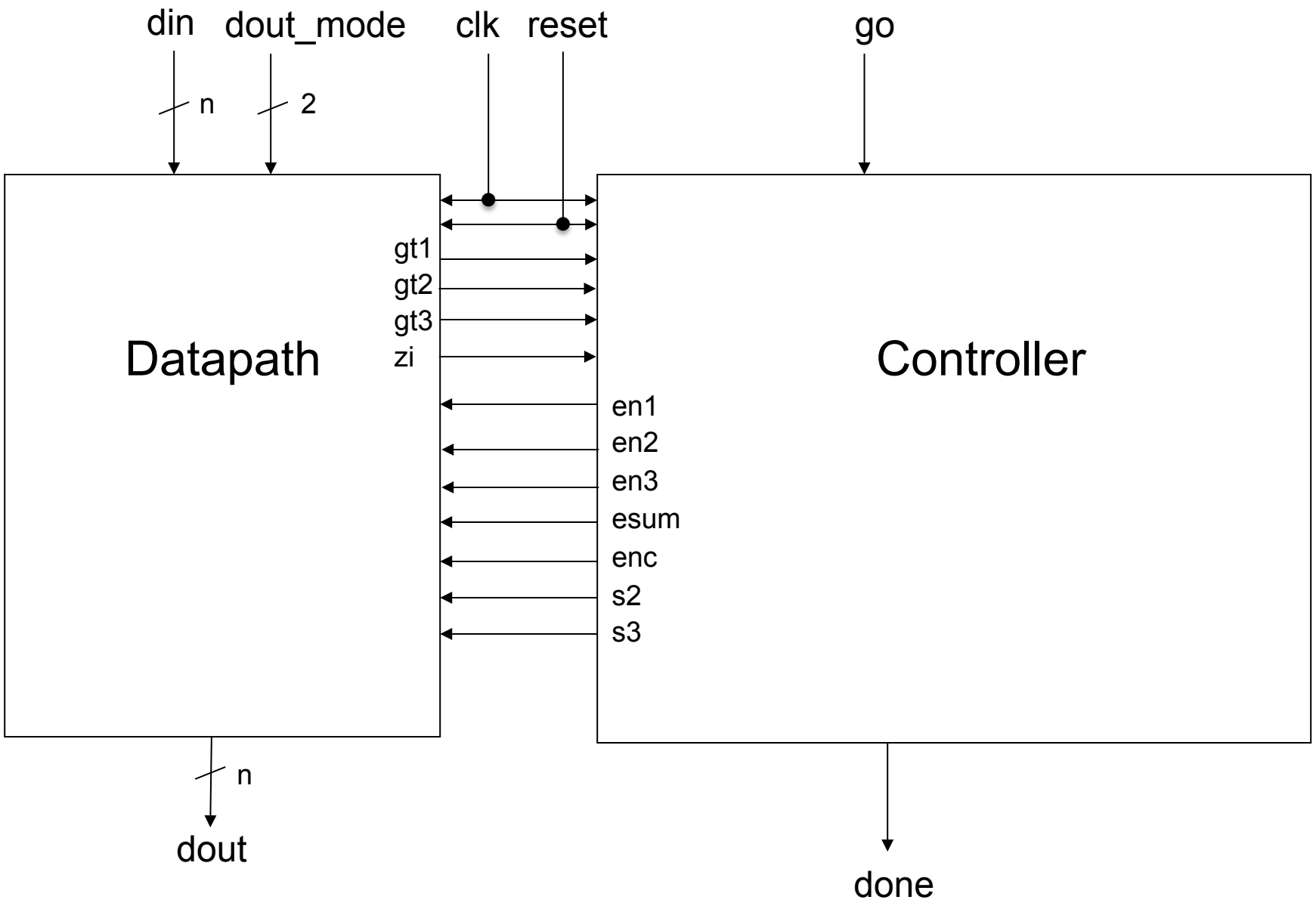


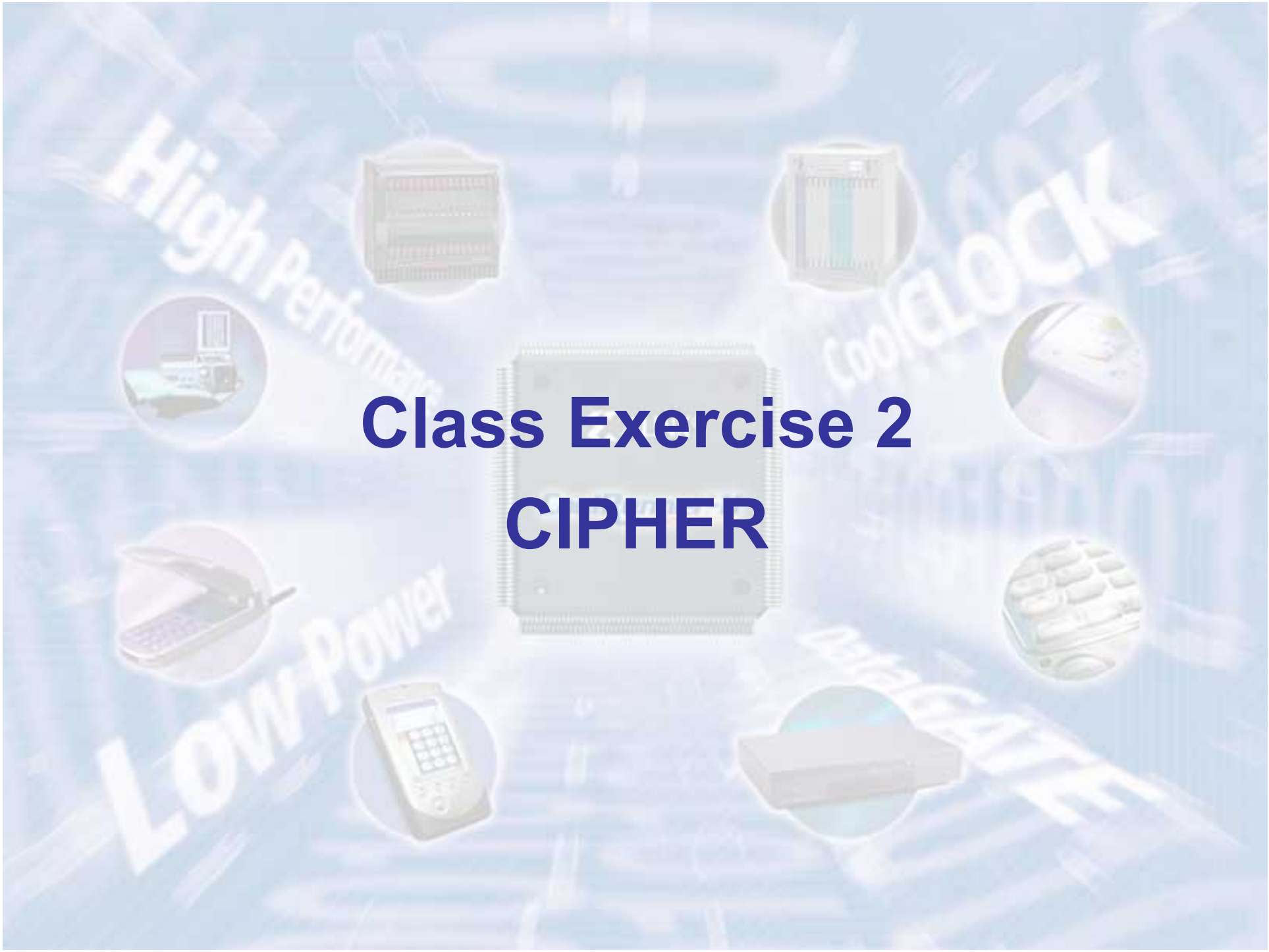
STATISTICS:
Solutions



Block diagram of the Datapath

Interface with the division into the Datapath and the Controller





Class Exercise 2

CIPHER

Pseudocode

Split input I into four words, I3, I2, I1, I0, of the size of w bits each

```
A = I3; B = I2; C = I1; D=I0
B = B + S[0]
D = D + S[1]
        for i = 1 to r do
        {
                T = (B*(2B + 1)) <<< k
                U = (D*(2D + 1)) <<< k
                A = ((A ⊕ T) <<< U) + S[2i]
                C = ((C ⊕ U) <<< T) + S[2i + 1]
                (A, B, C, D)  =  (B, C, D, A)
        }
A = A + S[2r + 2]
C = C + S[2r + 3]
O = (A, B, C, D)
```

Notation

w: word size, e.g., w=8 (constant)

k: $\log_2(w)$ (constant)

A, B, C, D, U, T: w-bit variables

I3, I2, I1, I0: Four w-bit words of the input I

r: number of rounds (constant)

O: output of the size of 4w bits

S[j] : 2r+4 round keys stored in two RAMs.

Each key is a w-bit word.

The first RAM stores values of S[j=2i], i.e., only round keys with even indices. The second memory stores values of S[j=2i+1], i.e., only round keys with odd indices.

Operations

⊕ **:** XOR

+ : addition modulo 2^w

– : subtraction modulo 2^w

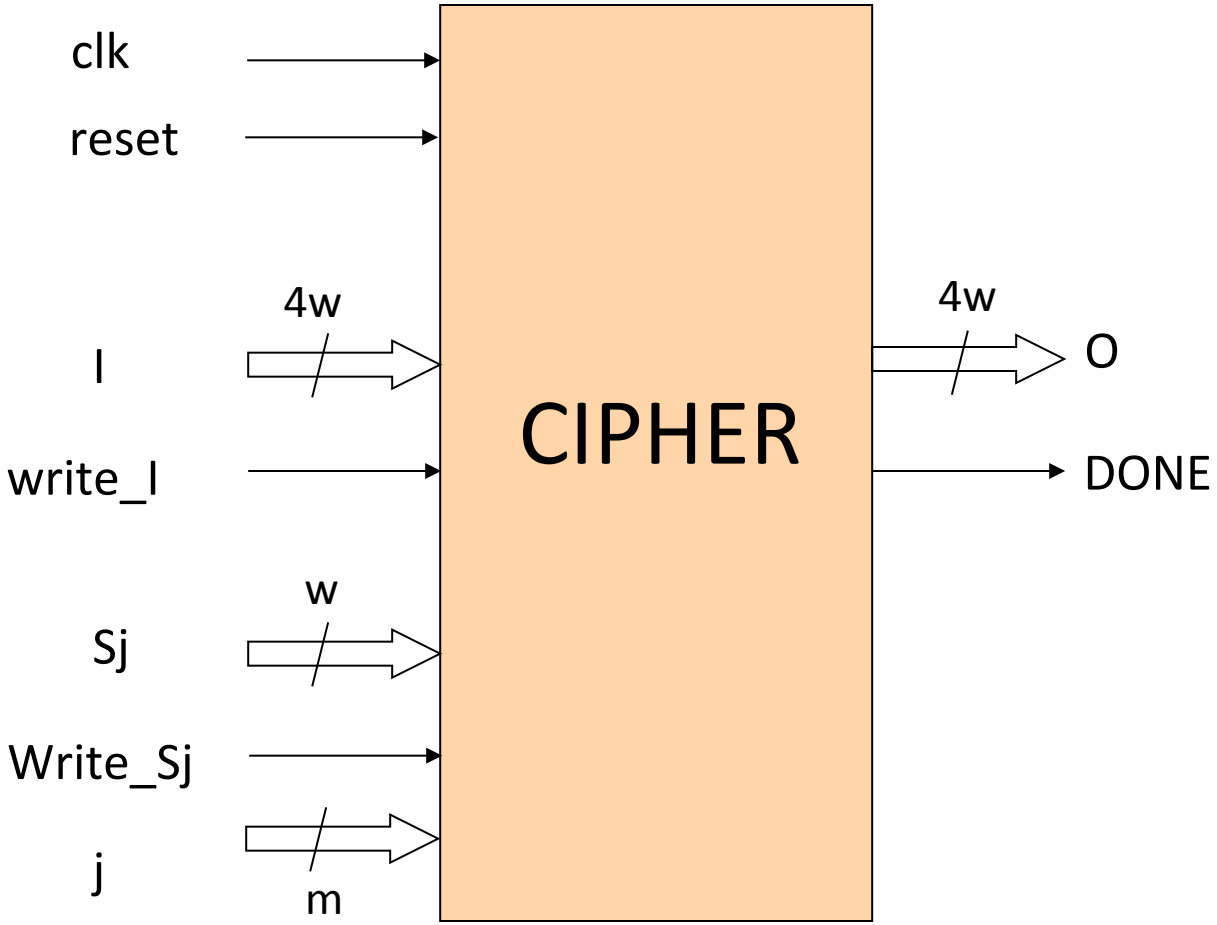
*** :** multiplication modulo 2^w

X <<< Y : rotation of X to the left by the number of positions given in Y

X >>> Y : rotation of X to the right by the number of positions given in Y

(A, B, C, D) : Concatenation of A, B, C, and D.

Circuit Interface



Interface Table

Port	Width	Meaning
clk	1	System clock.
reset	1	System reset – clears internal registers.
I	4w	Input block.
write_I	1	Synchronous write control signal for the input block I. After the block I is written to the CIPHER unit, the encryption of I starts automatically.
Sj	w	Round key S[j] loaded to one of the two internal memories.
write_Sj	1	Synchronous write control signal for the round key S[j].
j	m	Index of the round key S[j] loaded using input Sj.
O	4w	Output block.
DONE	1	Asserted when the output O is ready.

Note:

m is a size of index j. It is a minimum integer, such that $2^m - 1 \geq 2r+3$.

Protocol (1)

An external circuit first loads all round keys
S[0], S[1], S[2], …, S[2r+2], [2r+3]
to the two internal memories of the CIPHER unit.

The first memory stores values of S[j=2i], i.e., only round keys with even indices. The second memory stores values of S[j=2i+1], i.e. only round keys with odd indices.

Loading round keys is performed using inputs: Sj, j, write_Sj, clk.

Then, the external circuits, loads an input block I to the CIPHER unit, using inputs: I, write_I, clk.

After the input block I is loaded to the CIPHER unit, the encryption starts automatically.

Protocol (2)

When the encryption is completed, signal DONE becomes active, and the output O changes to the new value of the ciphertext.

The output O keeps the last value of the ciphertext at the output, until the next encryption is completed. Before the first encryption is completed, this output should be equal to zero.

Assumptions

- 2r+4 clock cycles are used to load round keys to internal RAMs
- one round of the main for loop of the pseudocode executes in one clock cycle
- you can access only one position of each internal memory of round keys per clock cycle

As a result, the encryption of a single input block I

should last r+2 clock cycles.