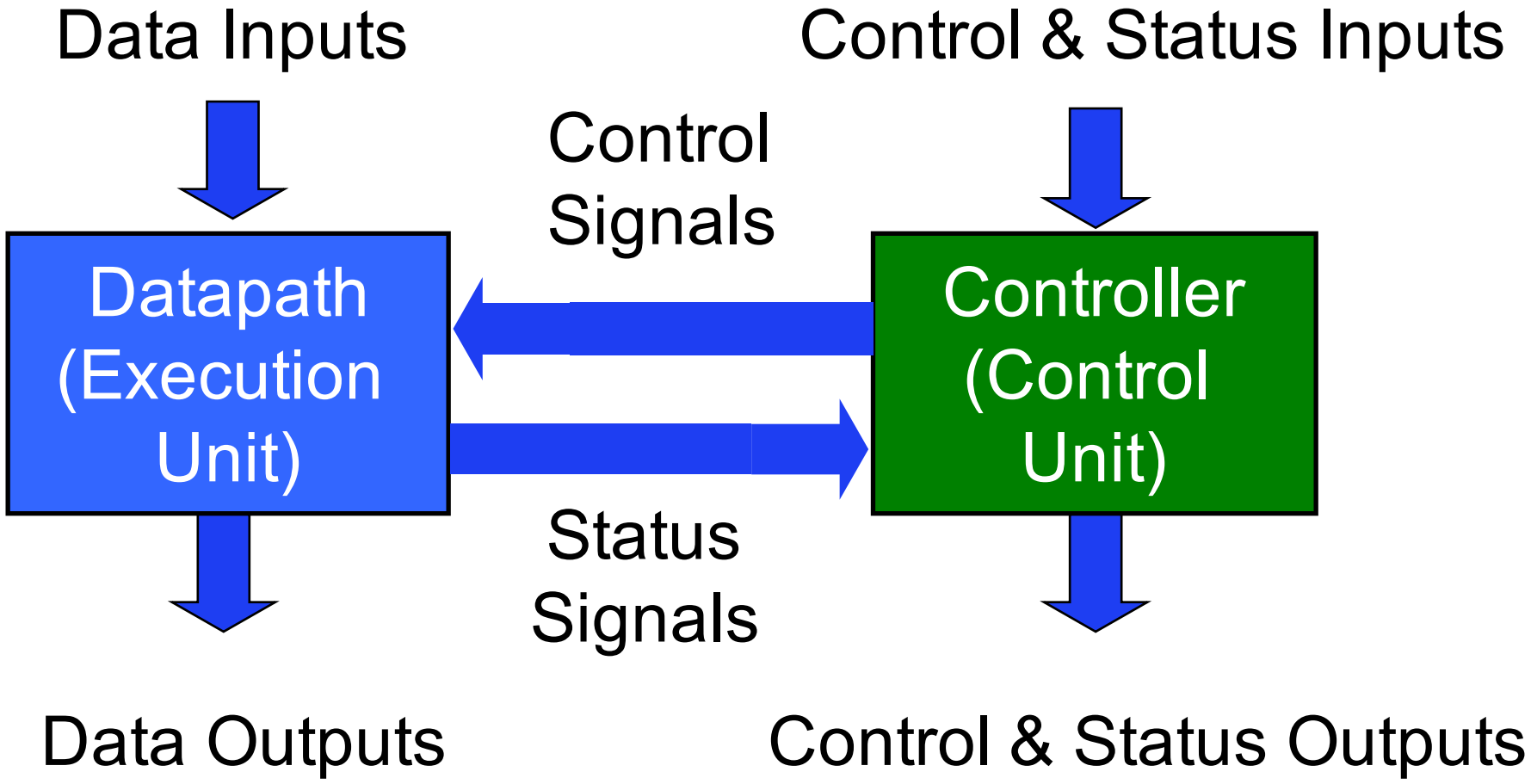
Lecture 5B

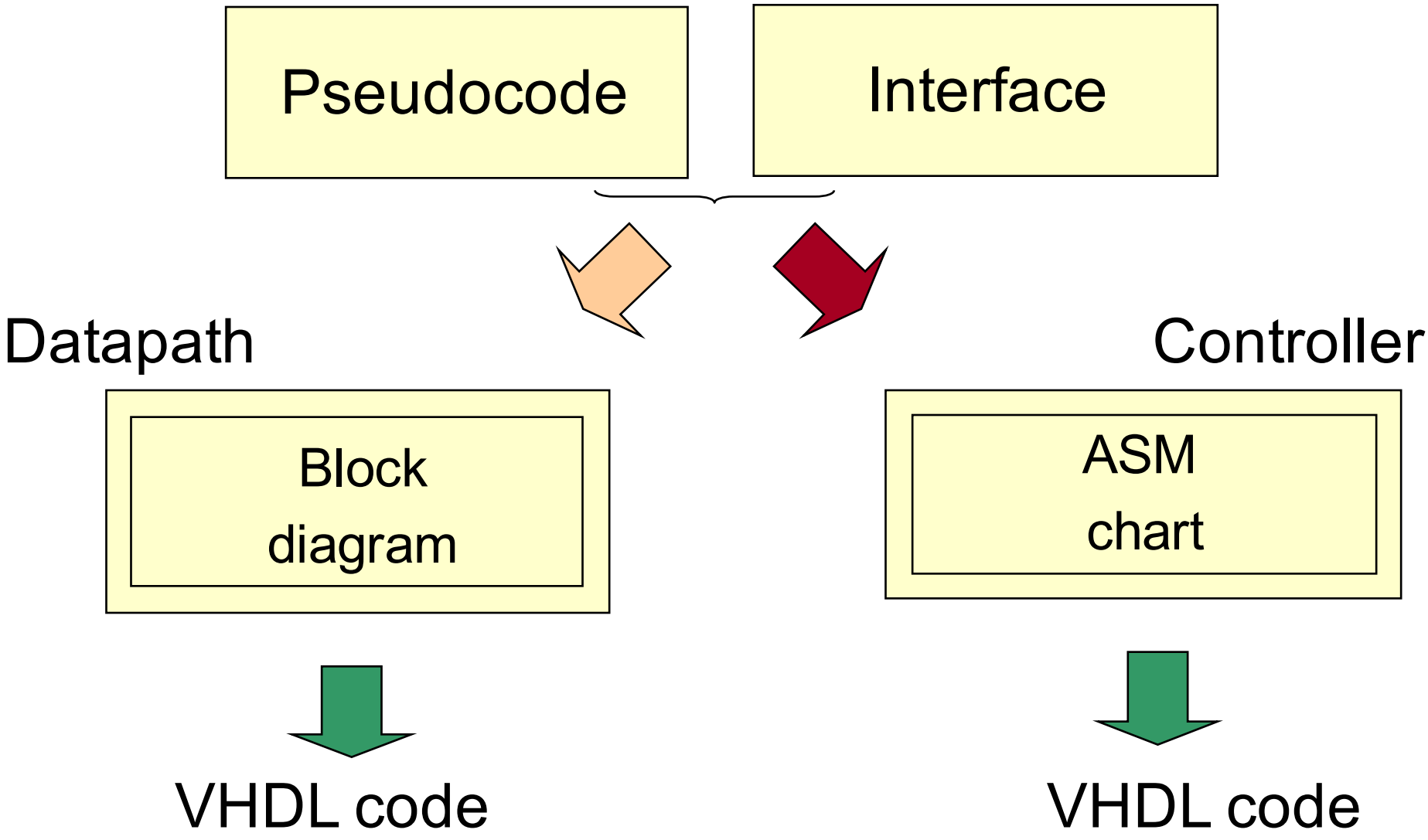
Block Diagrams

HASH Example

Structure of a Typical Digital System

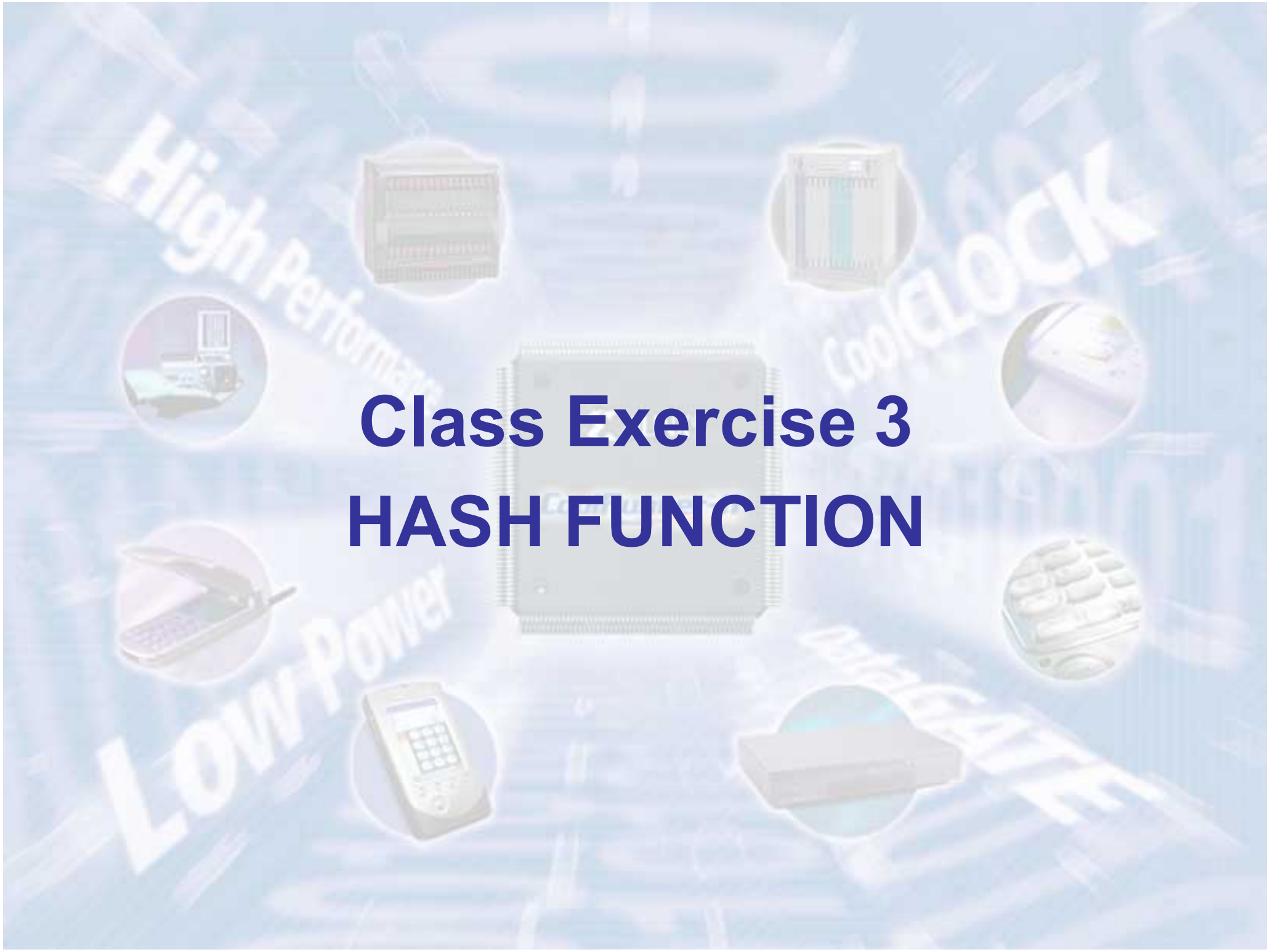


Hardware Design with RTL VHDL



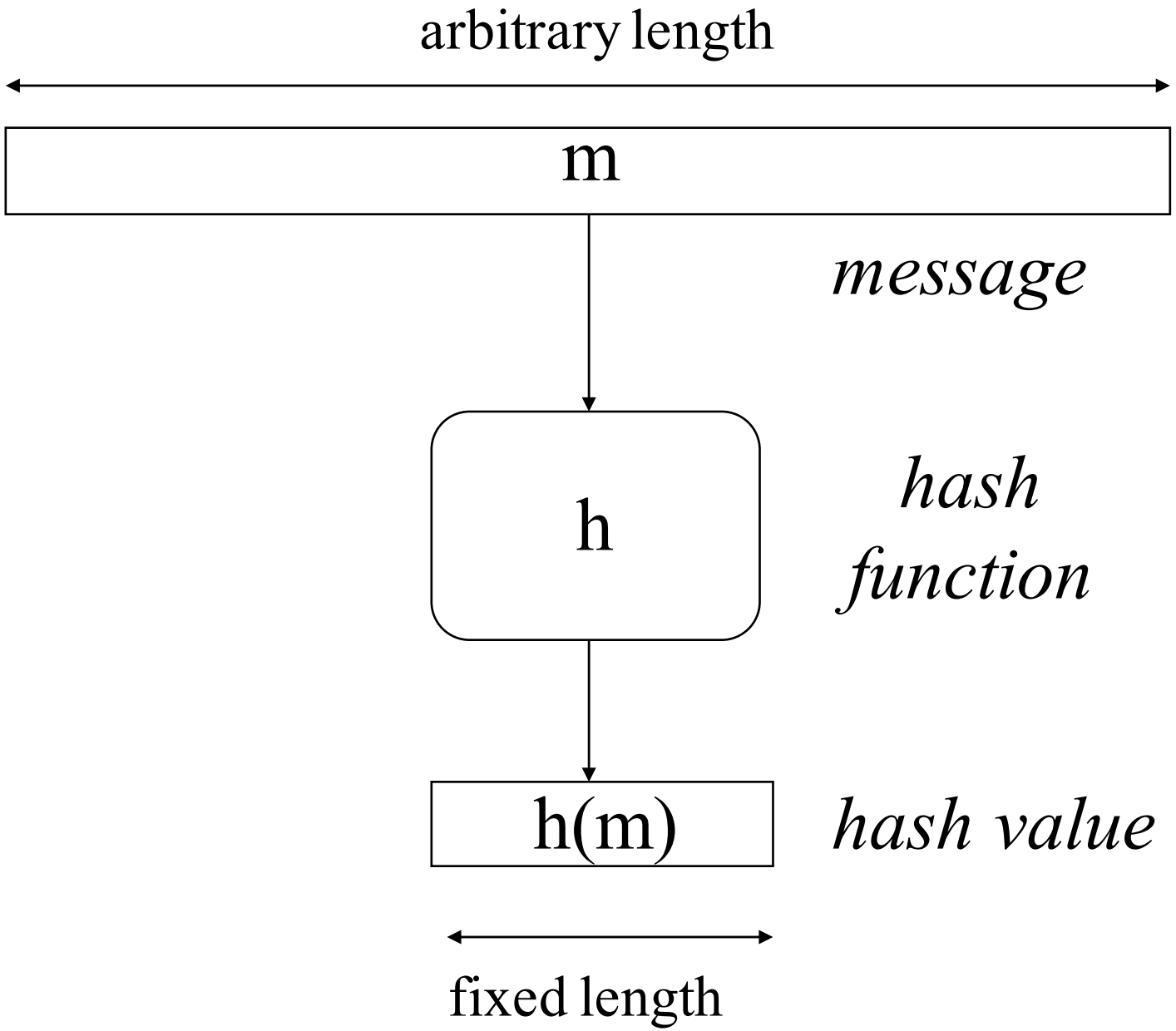
Steps of the Design Process Introduced in Class Today

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



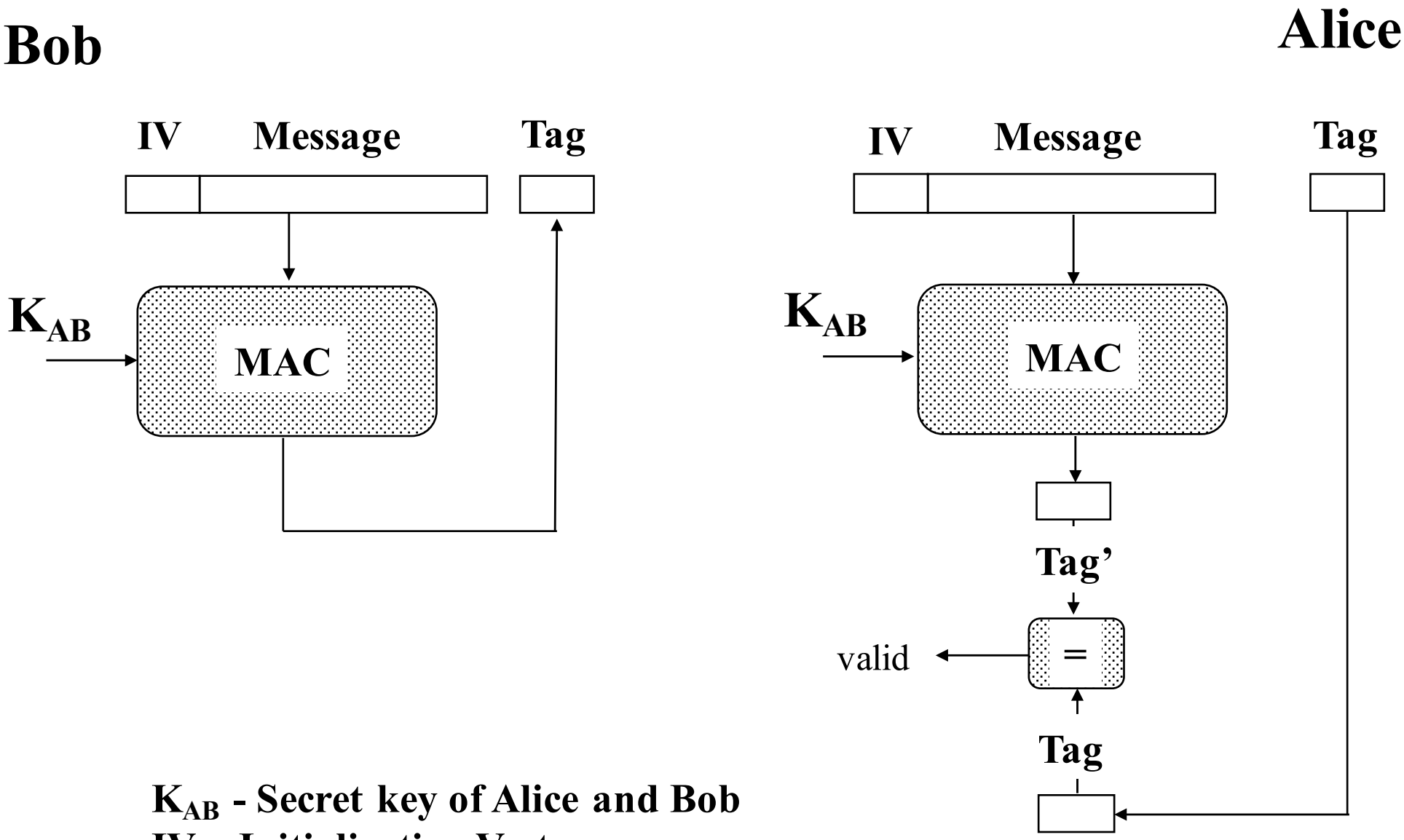
Class Exercise 3
HASH FUNCTION

Hash function



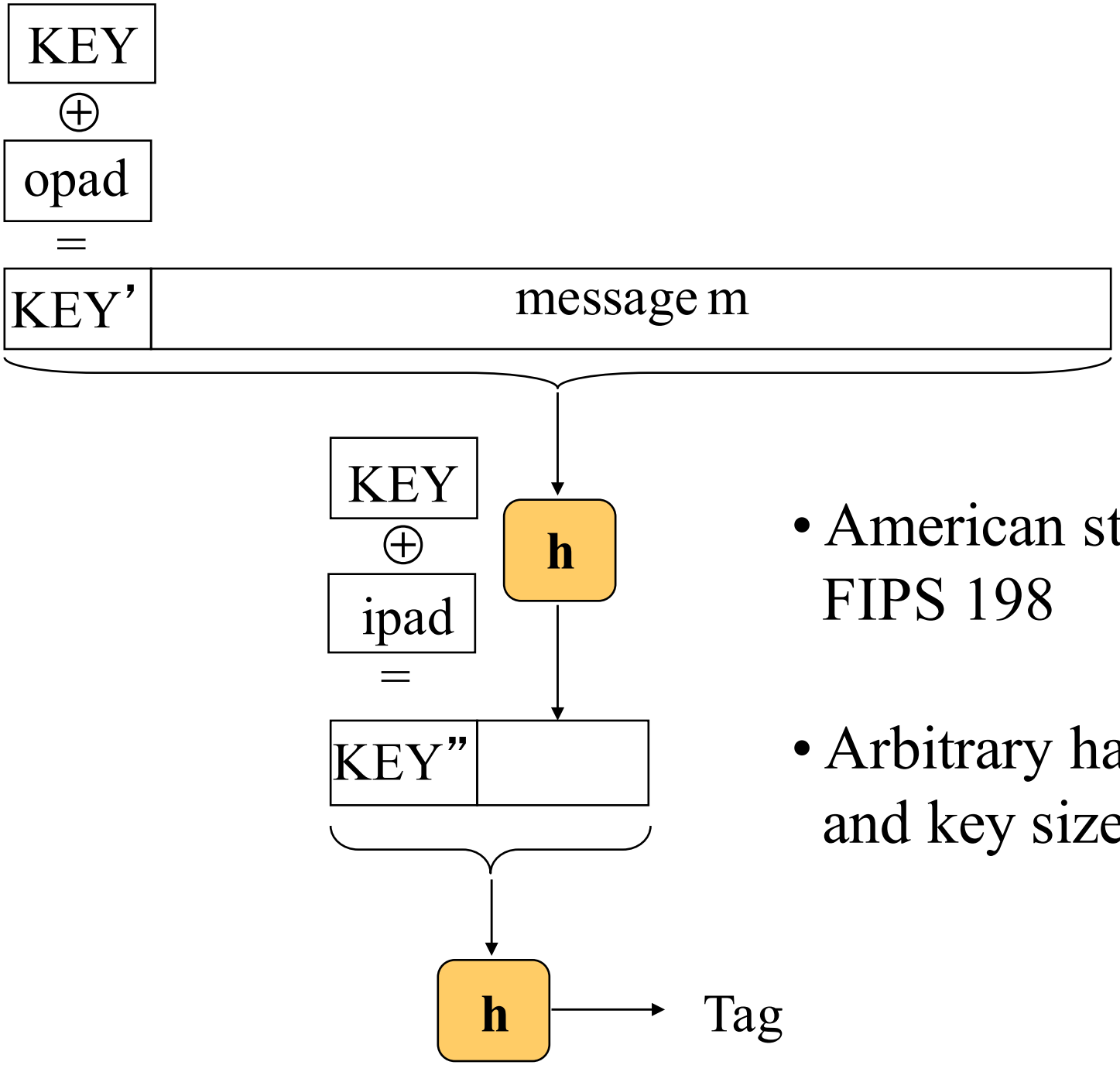
Authentication

Message Authentication Code - MAC



K_{AB} - Secret key of Alice and Bob
IV – Initialization Vector

HMAC



- American standard FIPS 198
- Arbitrary hash function and key size

Pseudocode (1)

```
h0 := iv0
h1 := iv1
h2 := iv2
h3 := iv3
last_block_stored :=0

while (last_block_stored != 1)
do
    wait until src_ready
    r := m
    last_block_stored := last_block

    a := h0
    b := h1
    c := h2
    d := h3
```

Pseudocode (2)

```
for i = 0 to 63 do
        ki := k[i mod 16]
        ri := r[i mod 4]
        f := (17*ri) mod 2^8
        e := (2d+1)*f mod 2^8
        d := (4c+2)*c mod 2^8
        c := b <<< 3
        b := a >>> i
        a := e ⊕ ki
    end for

    h0 := (h0 + a)  mod 2^8
    h1 := (h1 + b)  mod 2^8
    h2 := (h2 + c)  mod 2^8
    h3 := (h3 + d)  mod 2^8
end while

y := h0 || h1 || h2 || h3
done := 1
```

Notation

m: 32-bit message block (input)

r: 32-bit register

y: 32-bit circuit output (output)

iv0..iv3 : 8-bit initialization vectors (constants)

a..f, h0..h3, ki, ri : 8-bit intermediate values, treated as 8-bit unsigned integers

k[i]: 8-bit round constants: k[0]..k[15], stored in ROM

r[i]: bytes of the 32-bit register r, where

r[0] represents the least significant byte of r, and

r[3] represents the most significant byte of r.

Operations

⊕ : XOR

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

X || Y: X concatenated with Y

Interface Table

Port	Mode	Width	Function
clk	Input	1	System clock.
reset	Input	1	Asynchronous system reset.
m	Input	32	32-bit message block.
src_ready	Input	1	Control signal indicating that the source is ready. Must remain active until source is read.
src_read	Output	1	Control signal confirming that the source was read. Active for one clock cycle.
last_block	Input	1	Control signal indicating the last block of the message.
done	Output	1	Control signal indicating that the output is ready.
y	Output	32	Output y = h0 || h1 || h2 || h3.

Timing Requirements

Assume that

- one clock cycle is used for the once-per-message initialization:

  h0 = iv0; h1 = iv1; h2 = iv2; h3 = iv3;

- one clock cycle is used for the once-per-block initialization:

  a = h0; b = h1; c = h2; d = h3;

- one round of the main for-loop of the pseudocode executes in one clock cycle; there are a total of 64 rounds.

- one clock cycle is used for the once-per-block finalization:

  h0 = h0 + a; h1 = h1 + b; h2 = h2 + c; h3 = h3 + d;

Tasks

Draw a block diagram of the Datapath of the HASH circuit, using medium complexity components corresponding to the operations used in the pseudocode.

Clearly specify

- names, widths and directions of all buses
- names, widths and directions of all inputs and outputs of the logic components.

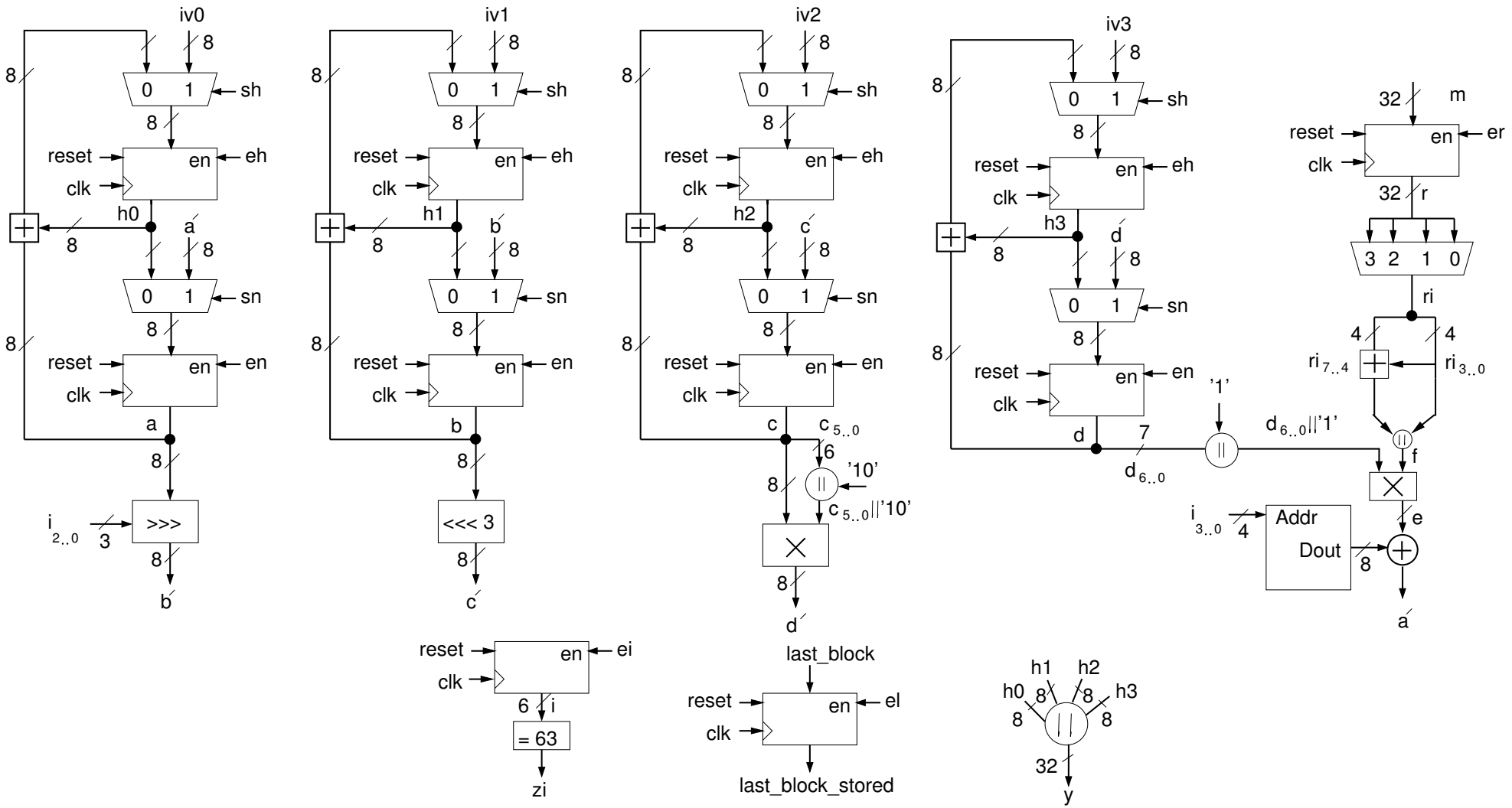
Assume that one round of the main for-loop of the pseudocode executes in ONE clock cycle.

Minimize the number of control signals to be generated by the Control Unit.

Mark the most likely critical path in your circuit.


HASH:
Solutions

Block Diagram



Interface with the Division into the Datapath and Controller

