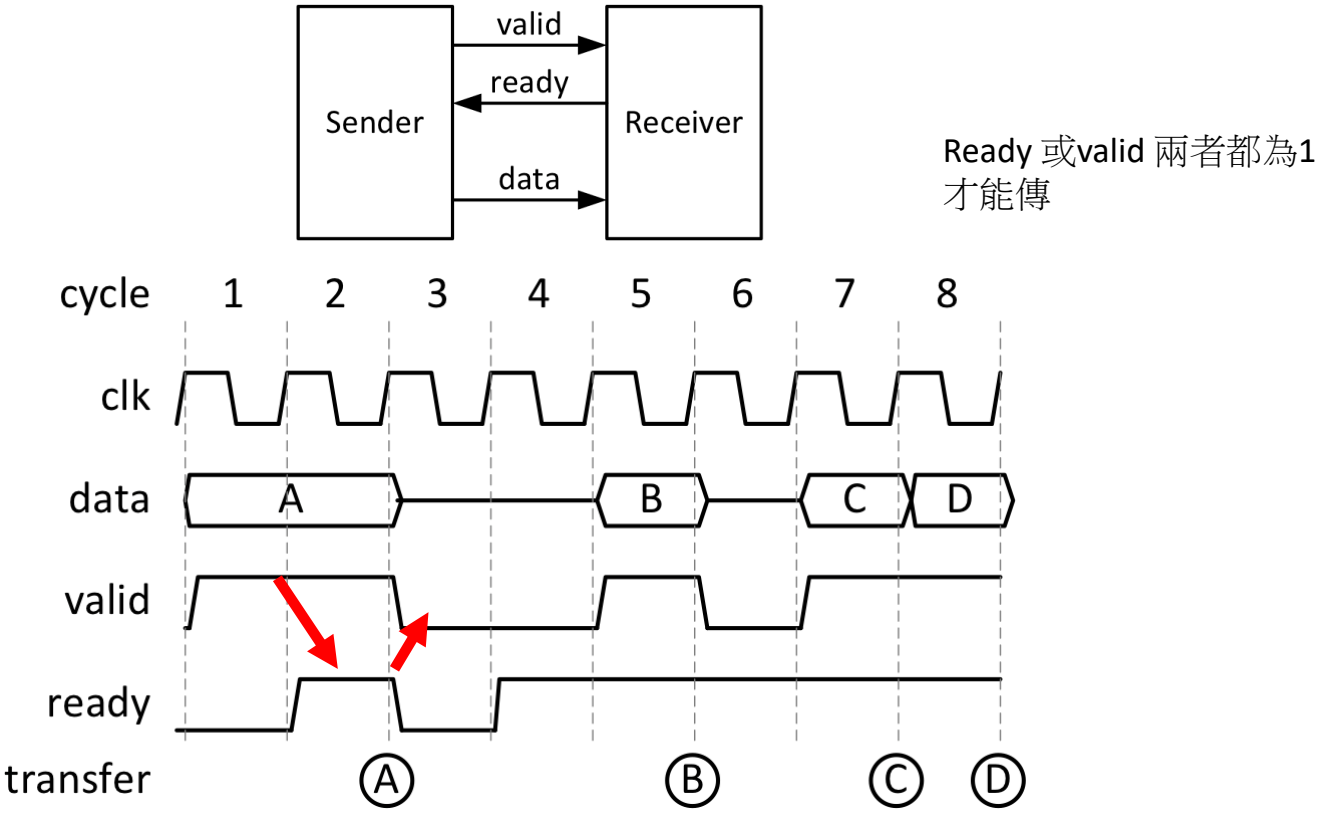
DCS Lab 5
AHB interconnect

Finite state machine

蔣洛亘

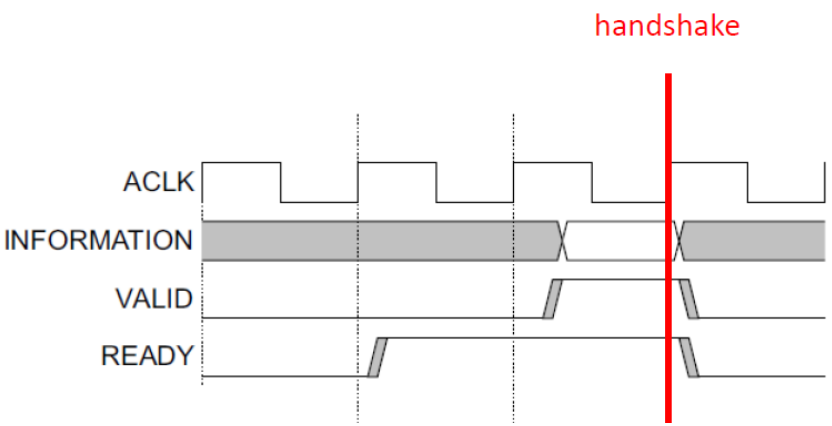
Flow Control: ready-valid flow

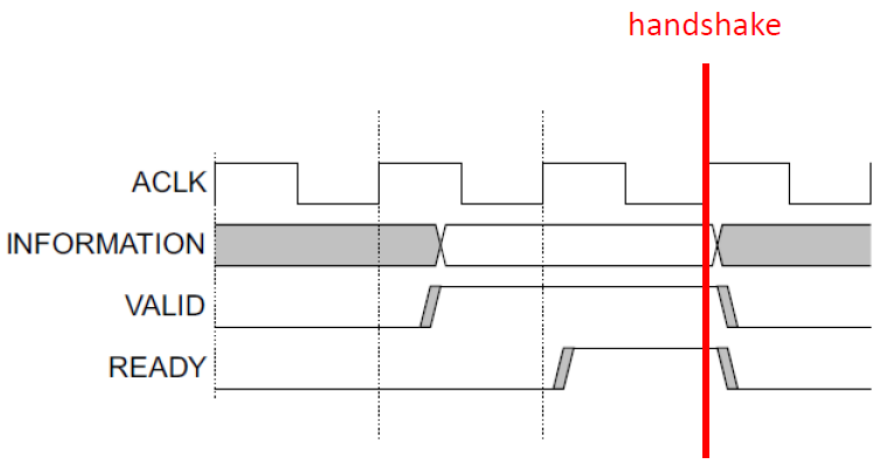


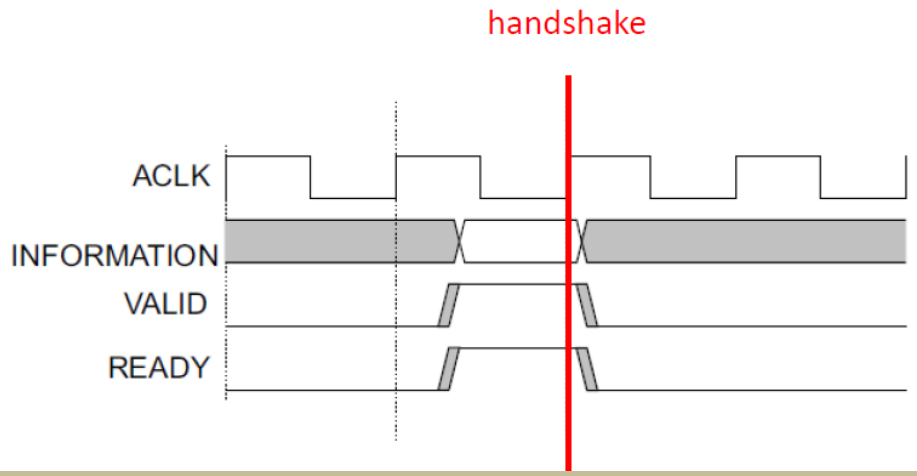
Ready 或valid 兩者都為1才能傳

Handshake process

- Ready before valid / Valid before ready
- Valid with ready

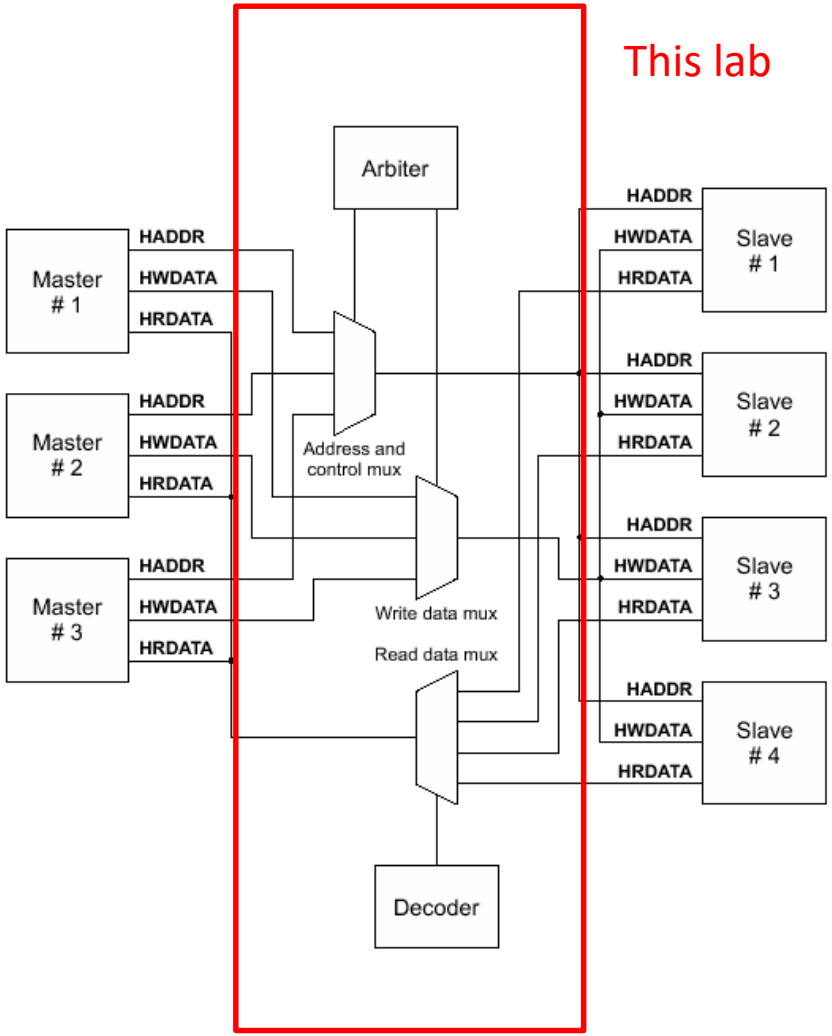






If(valid && ready)

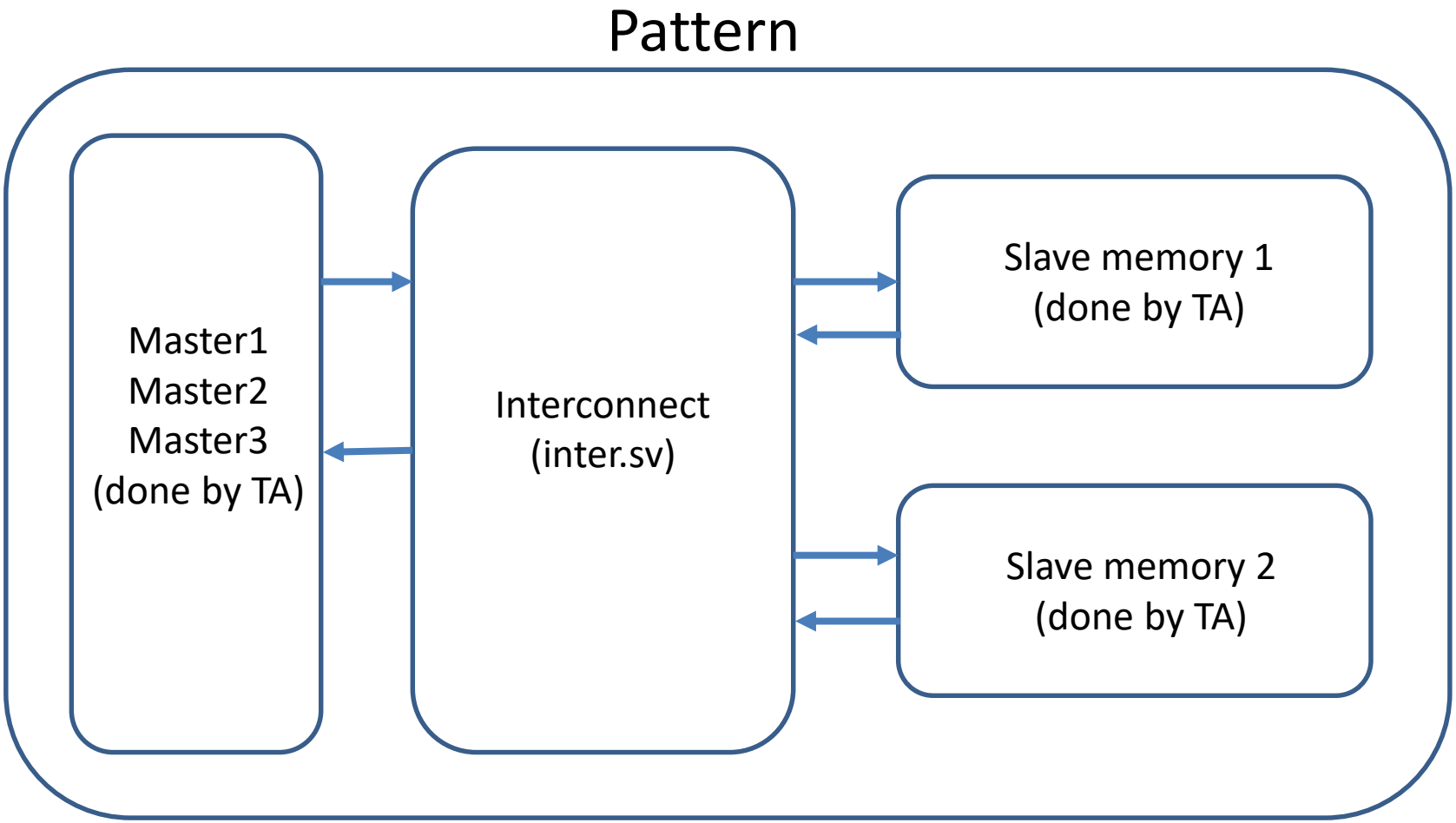
AHB Interconnect



Basic components:

- Master
- Slave
- Arbiter
- Decoder
- Mux

Lab - block diagram



Lab

- Masters will send input data to interconnect
- You should decode input data for valid, address and value
- Based on master priority(1->2->3), send data to slave memory
- Output handshake signal

Lab - decode

- [6:0]data_in_1 (from Master1)
- data_in_1[6]: 0->slave1 1->slave2
- data_in_1[5:3]: address
- data_in_1[2:0]: value
- Ex: 7'b0101001 for slave1, addr=5, value=1
- Ex: 7'b1011110 for slave2, addr=3, value=6

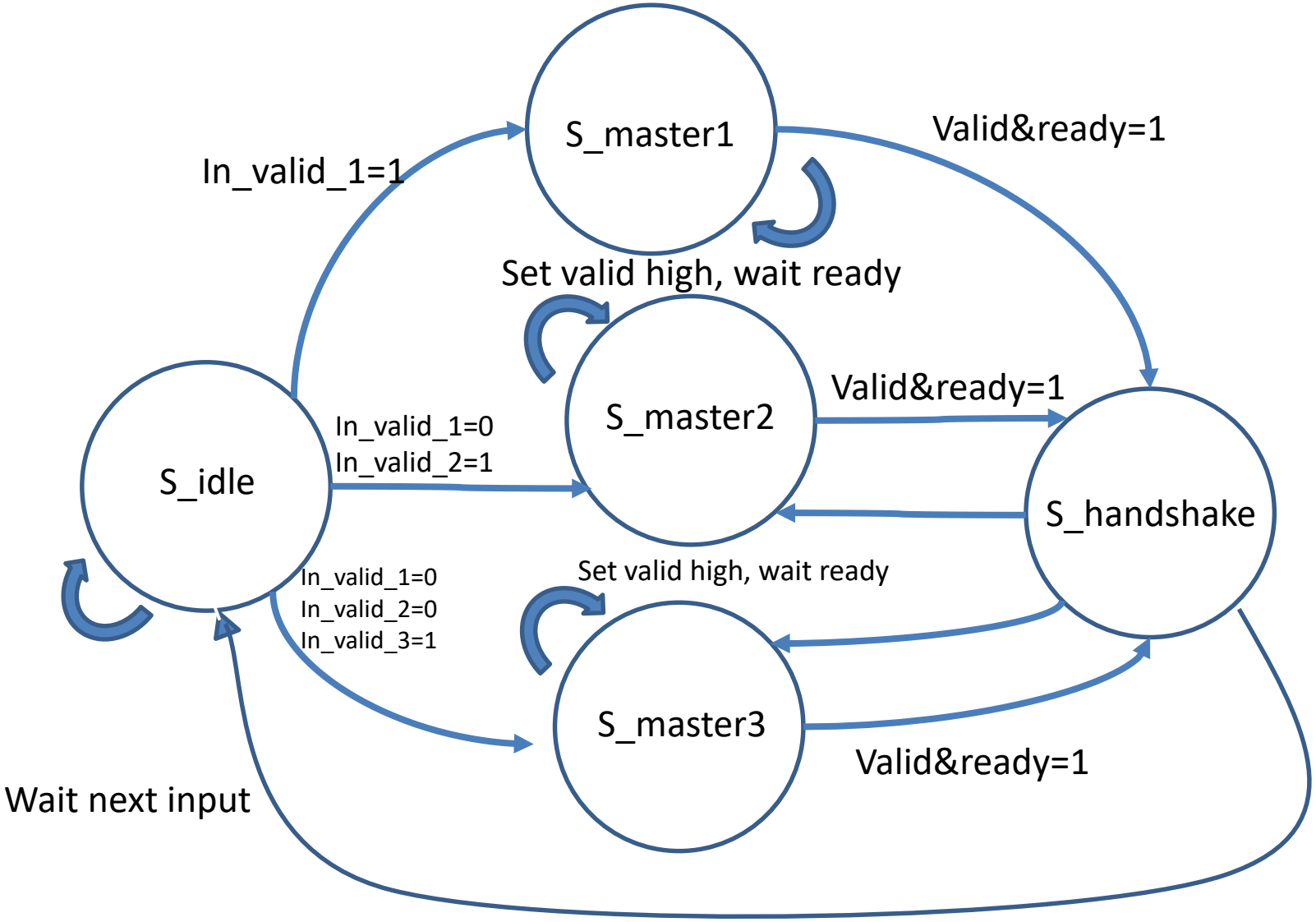
inter.sv

Input Signal	Bit Width	Definition
clk	1	Clock
rst_n	1	Asynchronous active-low reset
in_valid_1	1	in_valid from master1
in_valid_2	1	in_valid from master2
in_valid_3	1	in_valid from master3
data_in_1	7	Data from master1
data_in_2	7	Data from master2
data_in_3	7	Data from master3
ready_slave1	1	Ready signal from slave1
ready_slave2	1	Ready signal from slave2

inter.sv

Output Signal	Bit Width	Definition
valid_slave1	1	valid signal to slave1
valid_slave2	1	valid signal to slave2
addr_out	3	Address you want to write
value_out	3	Value you want to save
handshake_slave1	1	High for 1 cycle after handshake with slave1
handshake_slave2	1	High for 1 cycle after handshake with slave2

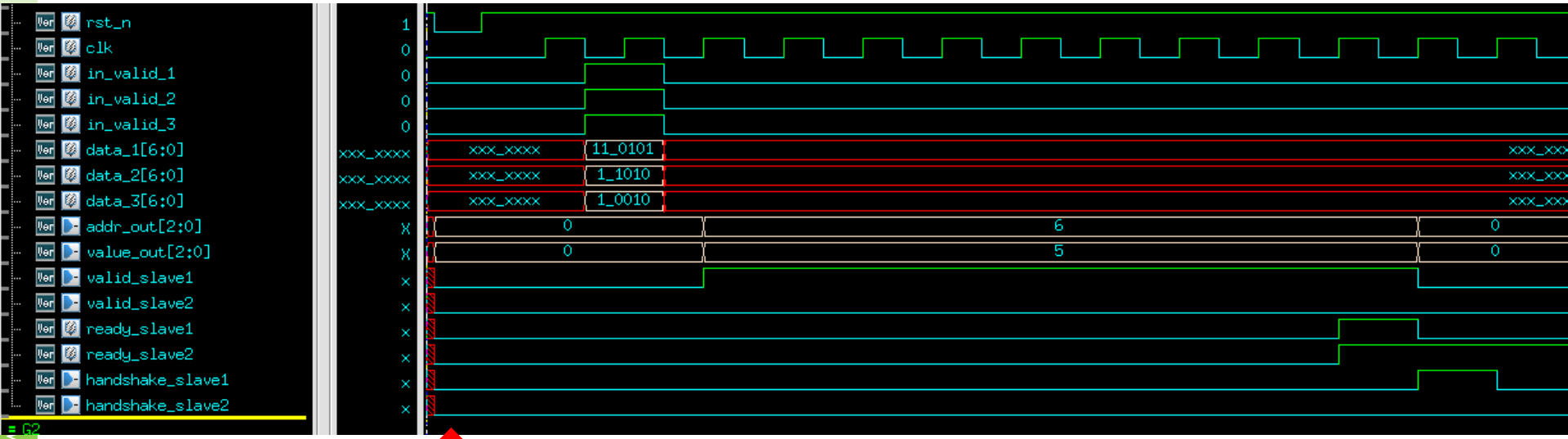
Lab – Arbiter FSM (ref.)



Spec

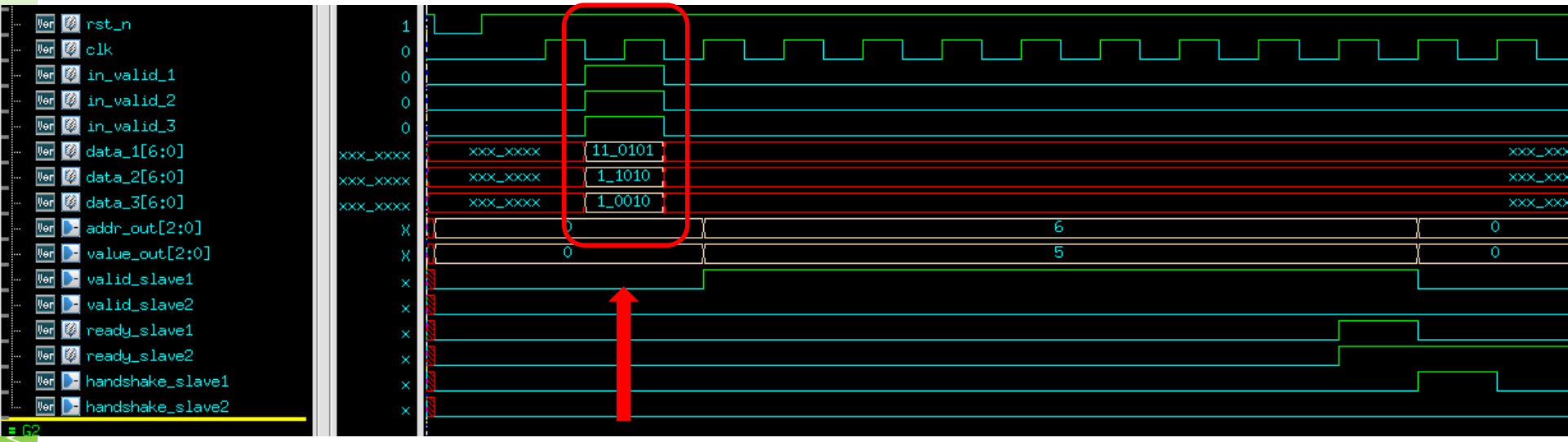
- 不可以超過30個cycles沒有完成handshake(從invalid或上一次handshake開始數)
- 拉起handshake signal(only 1 cycle)後,pattern會檢查全部slave memory的值，必須依照master priority
- 可參考助教提供的FSM。
- 所有output必須非同步負準位reset。
- 01_RTL 需要PASS。
- 02_SYN不能有error跟latches。
- 02_SYN時間timing slack必須為MET。
- 03_GATE 需要PASS

Waveform

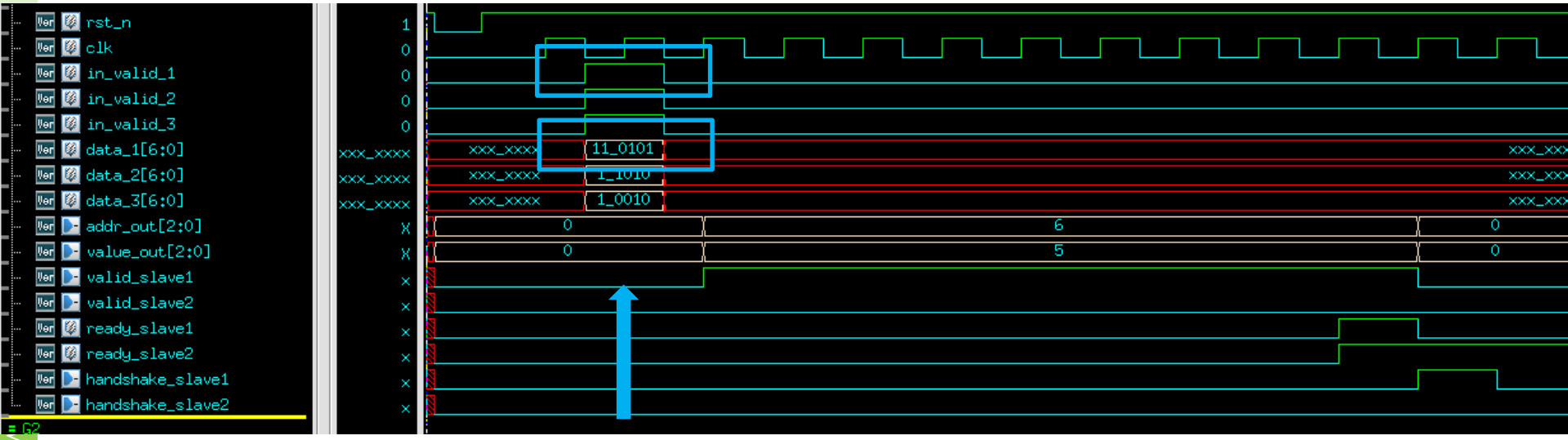


negative trigger asynchronous reset

Waveform


rst_n
clk
in_valid_1
in_valid_2
in_valid_3
data_1[6:0]
data_2[6:0]
data_3[6:0]
addr_out[2:0]
value_out[2:0]
valid_slave1
valid_slave2
ready_slave1
ready_slave2
handshake_slave1
handshake_slave2
xxx_xxxx
11_0101
1_1010
1_0010
Input data

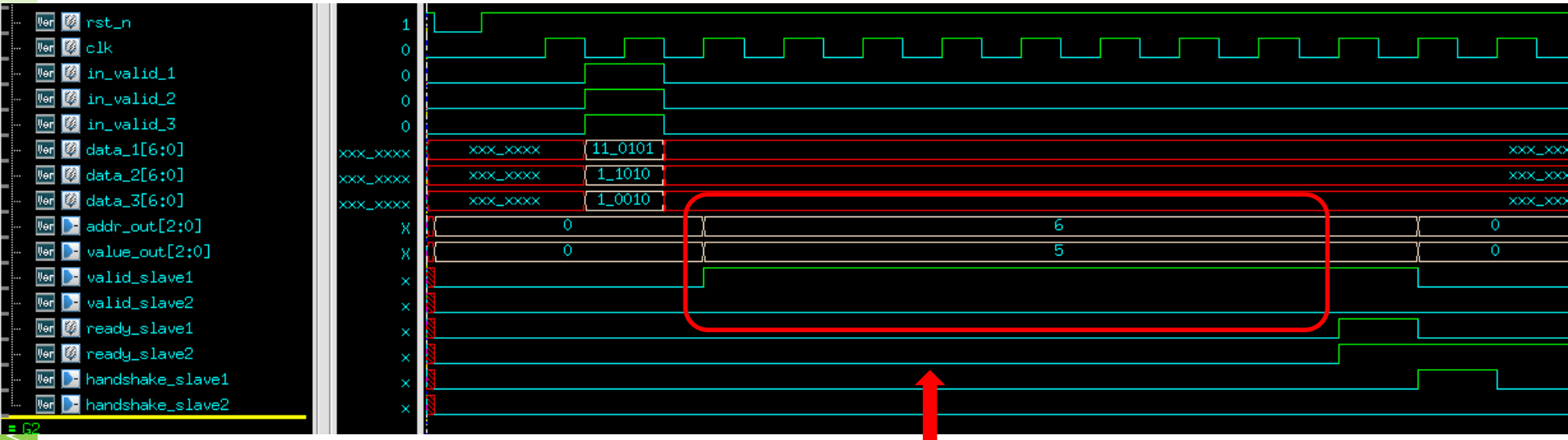
Waveform



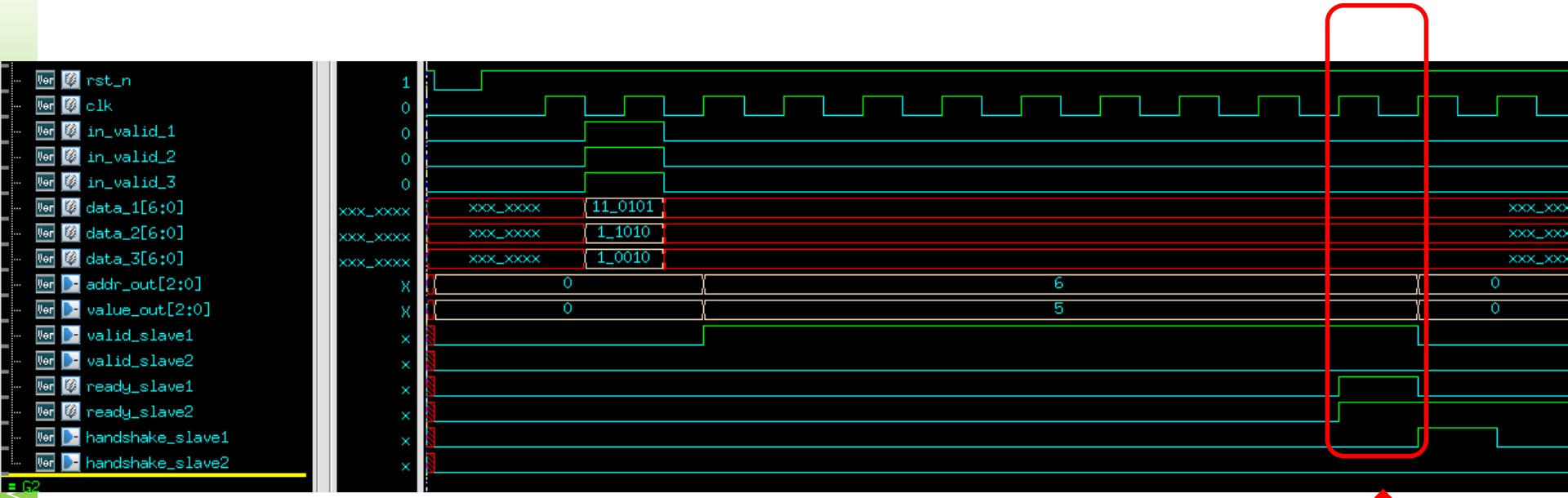
Master1 first

Decode: 0 110 101 => slave1 address:6 value:5

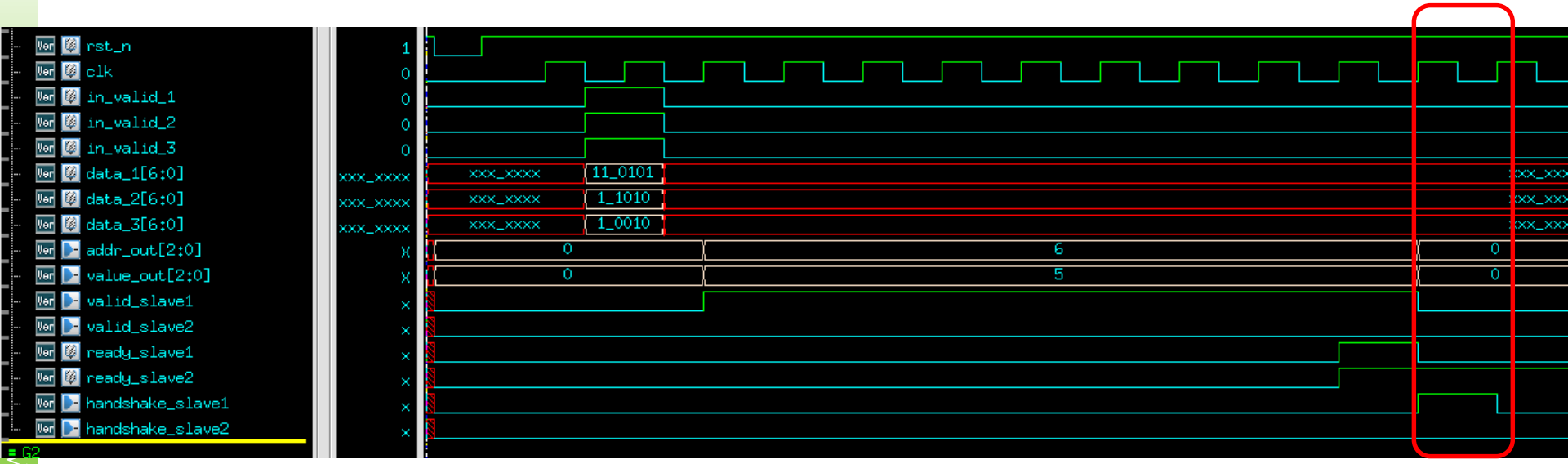
Waveform


rst_n
clk
in_valid_1
in_valid_2
in_valid_3
data_1[6:0]
data_2[6:0]
data_3[6:0]
addr_out[2:0]
value_out[2:0]
valid_slave1
valid_slave2
ready_slave1
ready_slave2
handshake_slave1
handshake_slave2
11_0101
1_1010
1_0010
Set valid high, wait slave1 ready

Waveform


rst_n
clk
in_valid_1
in_valid_2
in_valid_3
data_1[6:0]
data_2[6:0]
data_3[6:0]
addr_out[2:0]
value_out[2:0]
valid_slave1
valid_slave2
ready_slave1
ready_slave2
handshake_slave1
handshake_slave2
xxx_xxxx
11_0101
1_1010
1_0010
Handshake!

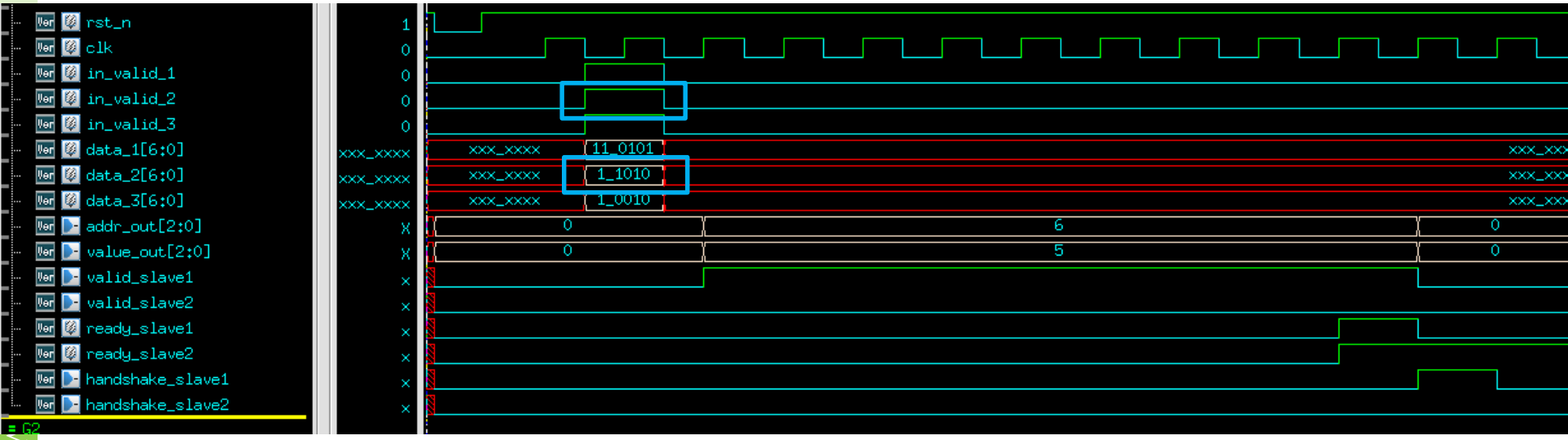
Waveform



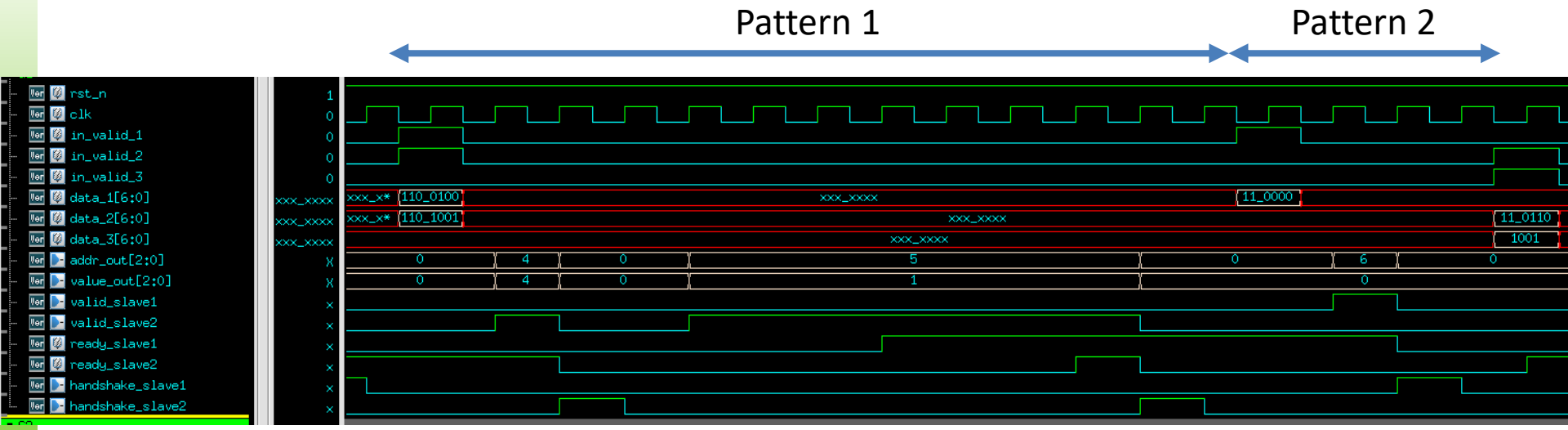
output handshake signal

Waveform

Continue for Master2



Waveform


Pattern 1
Pattern 2
rst_n
clk
in_valid_1
in_valid_2
in_valid_3
data_1[6:0]
data_2[6:0]
data_3[6:0]
addr_out[2:0]
value_out[2:0]
valid_slave1
valid_slave2
ready_slave1
ready_slave2
handshake_slave1
handshake_slave2
110_0100
110_1001
11_0000
11_0110
1001

Command

- tar -xvf ~dcsta01/Lab05.tar
- cd Lab05/01_RTL/
- Need 02_SYN
  - No Latch
  - No error
  - No timing violation (MET)
- Need 03_GATE
- Separate combinational and sequential blocks

Demo1: 3/31(四), 16:25:00

Demo2: 3/31(四), 23:59:59