Homework II 說明

Instructor：Lih-Yih Chiou

TA : Henry

Date : 10/11/2018

EE

LPHPLAB VLSI Design LAB

Outline

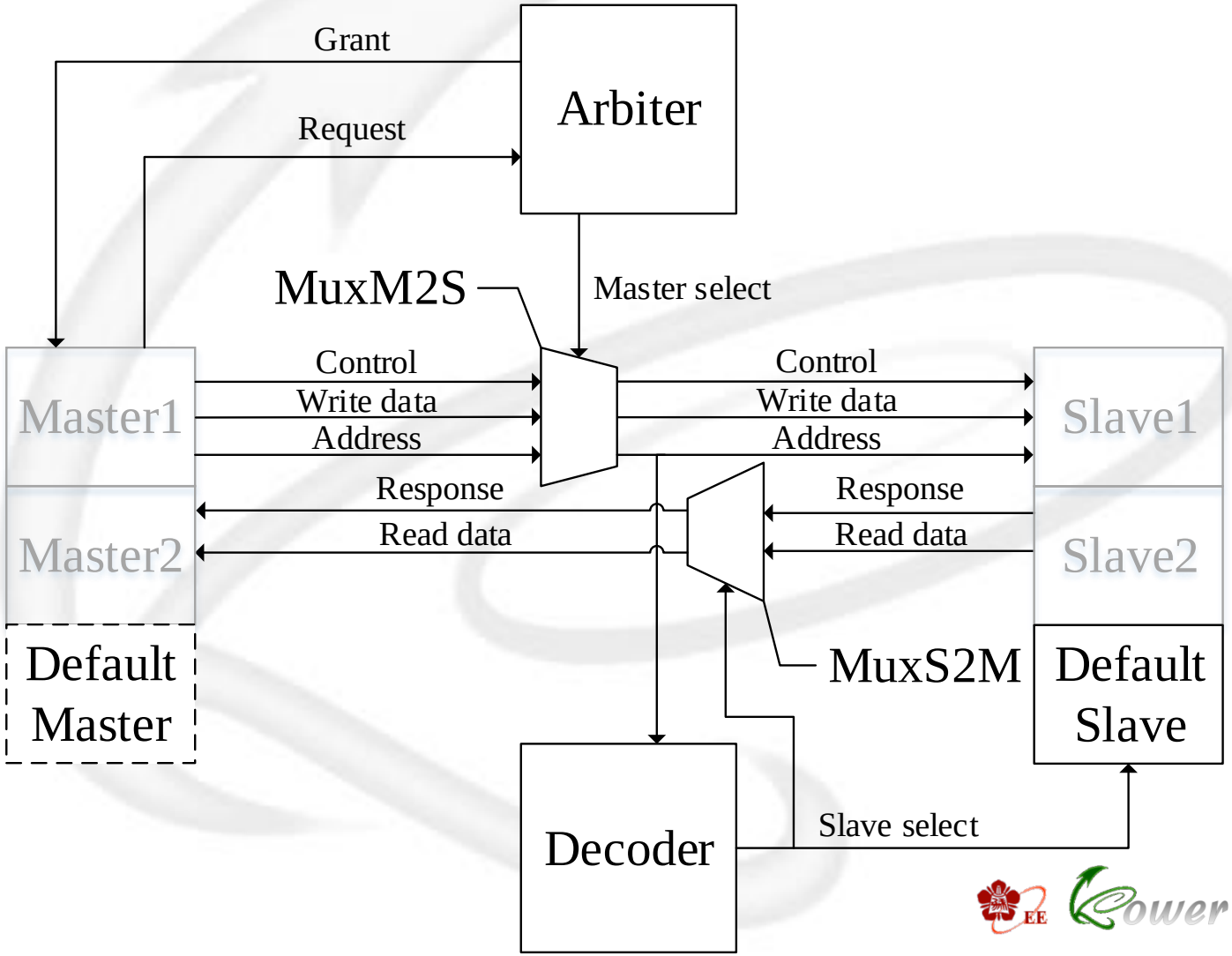
- Problem 1
  - 作業內容說明
  - 作業驗證說明
- Problem 2
  - 作業內容說明
  - 作業驗證說明
- 作業繳交注意事項

Problem 1

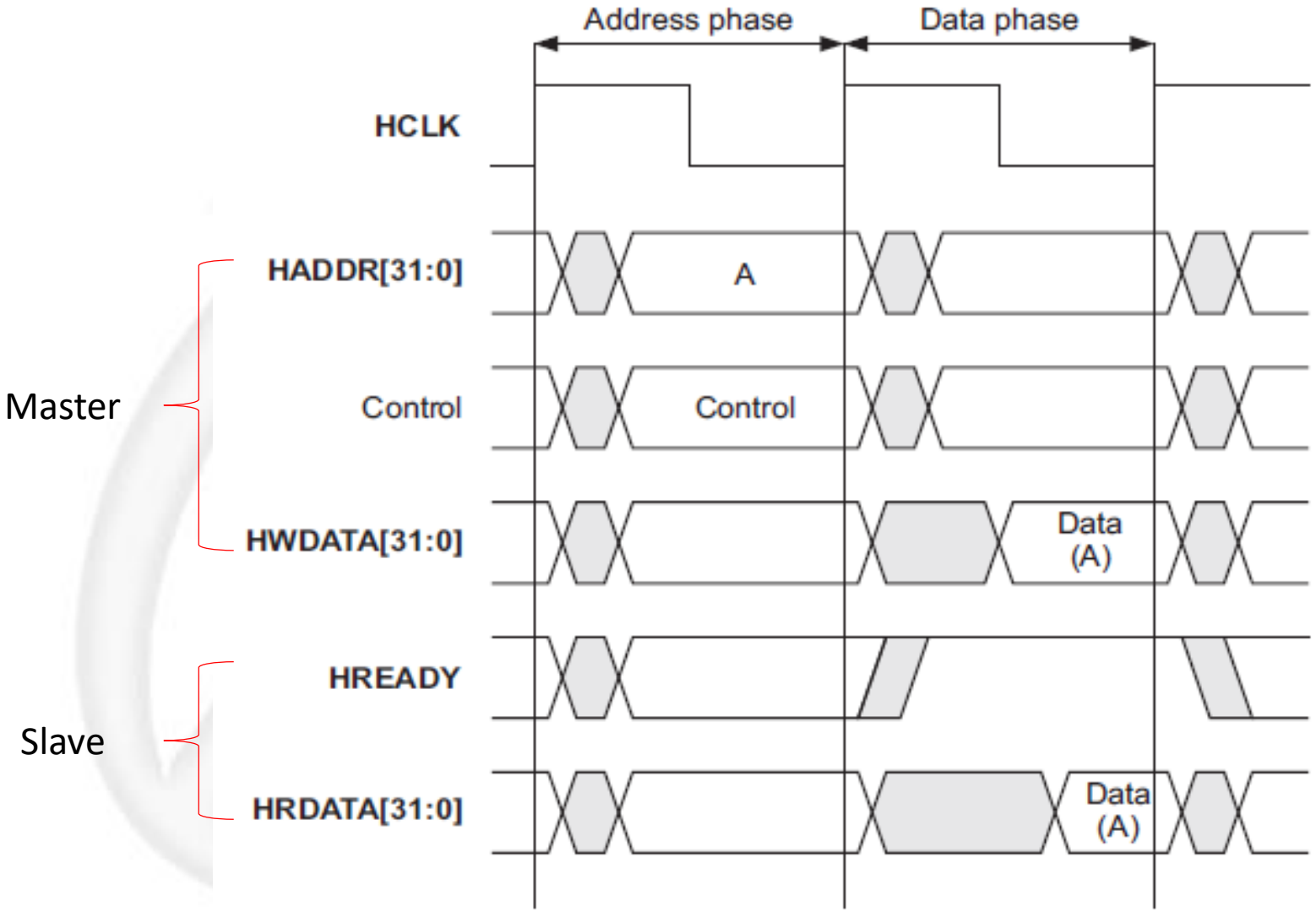
作業內容說明

Block Diagram

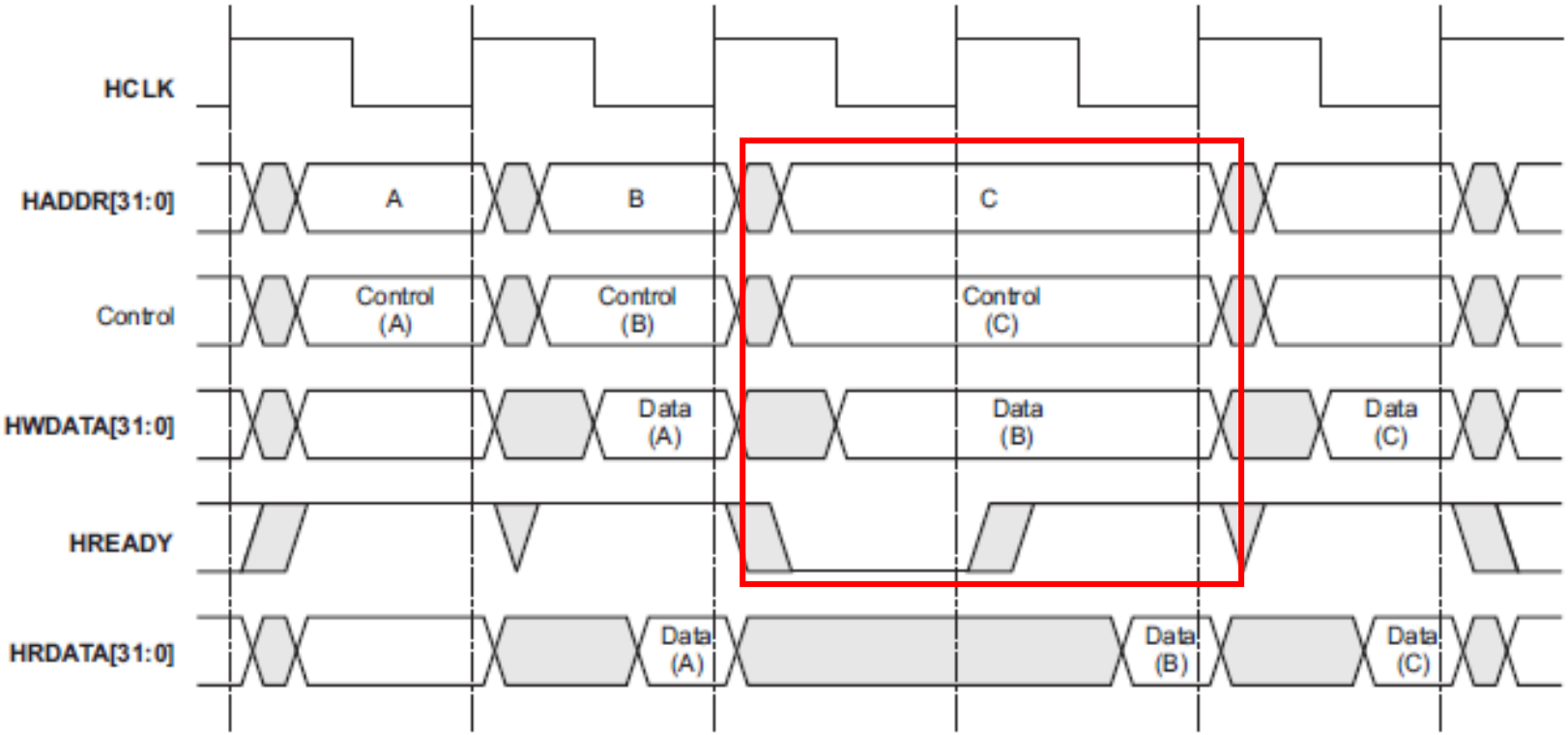
- 根據AMBA Specification 2.0完成AHB之設計



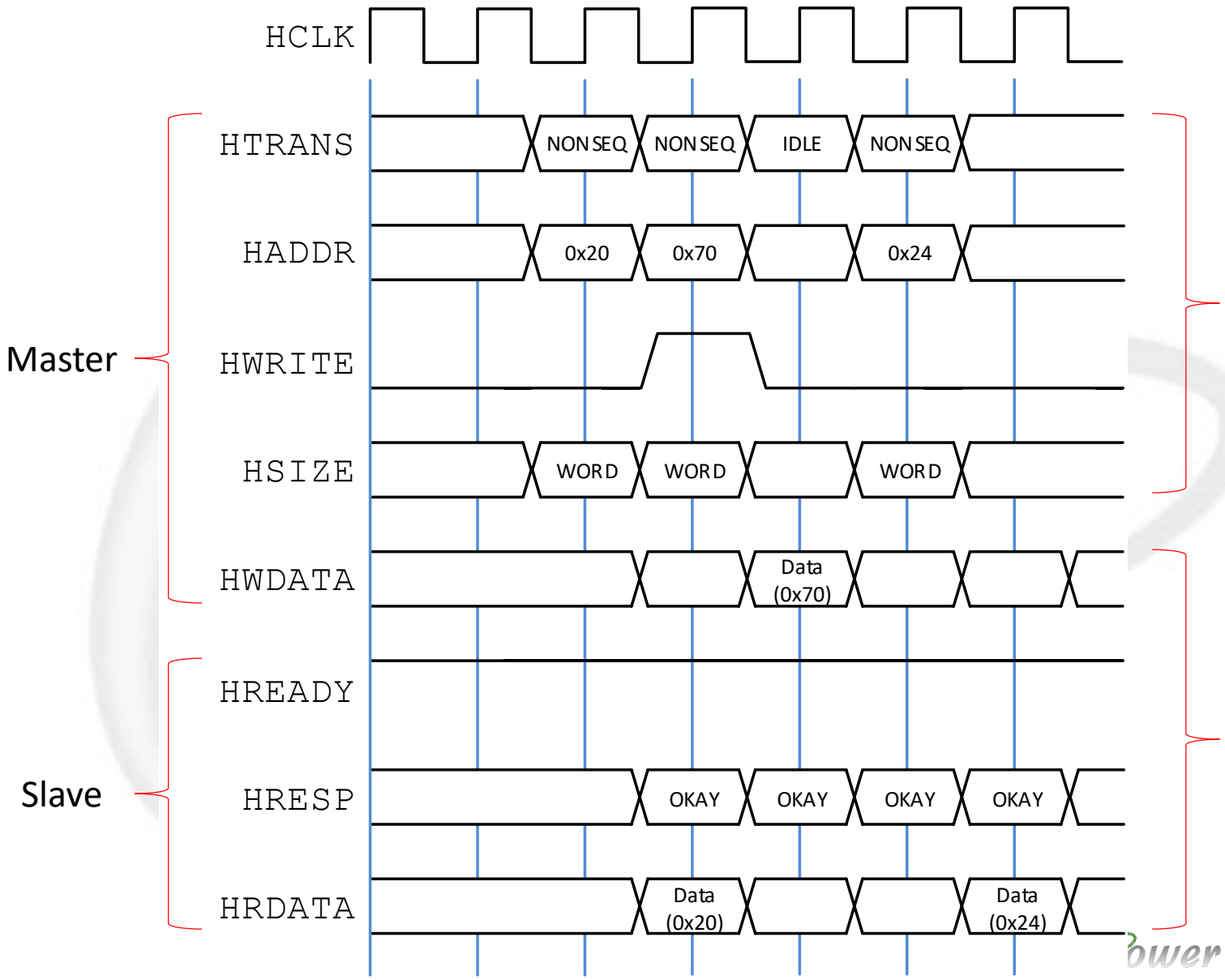
AHB Basic Transfer



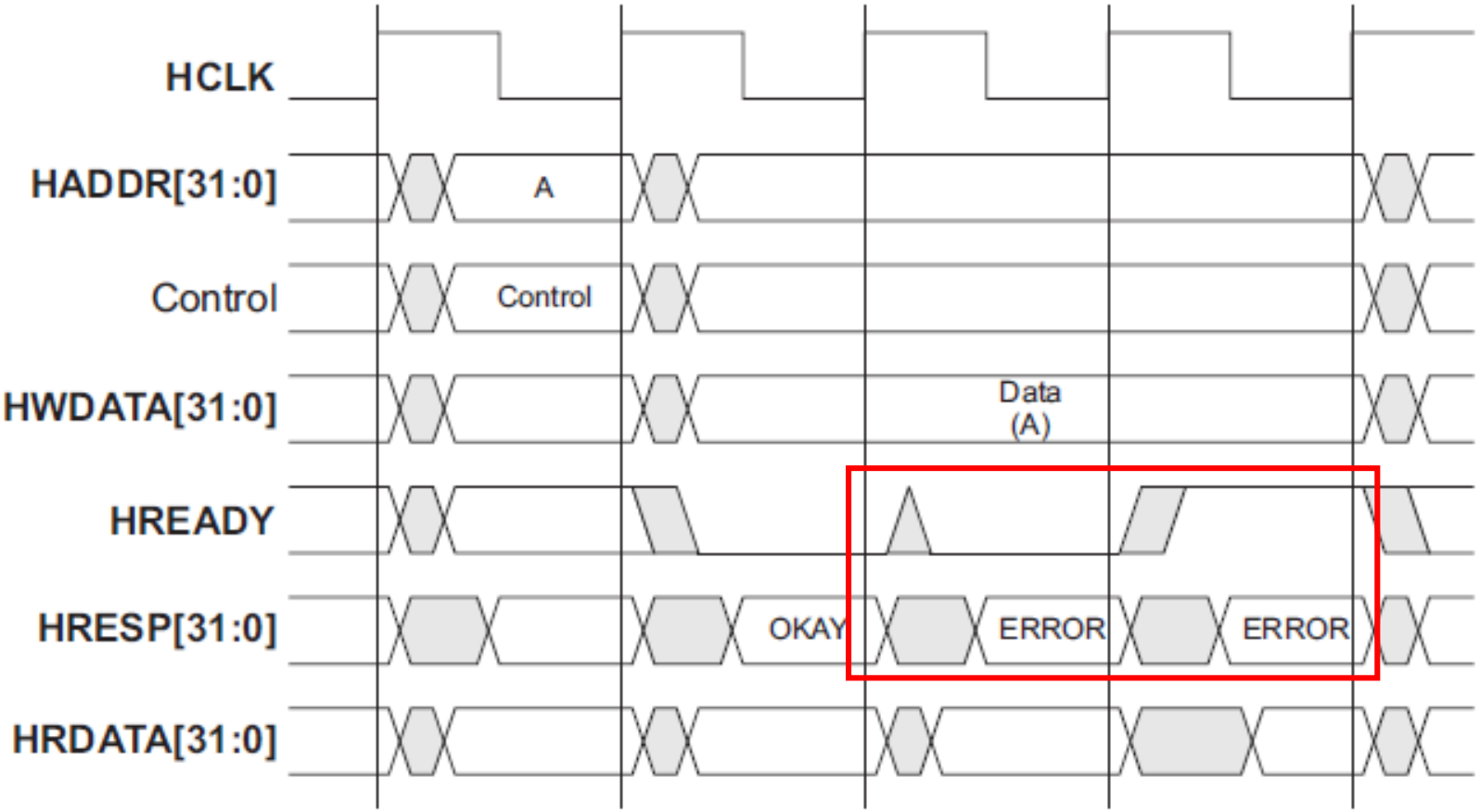
AHB Wait State



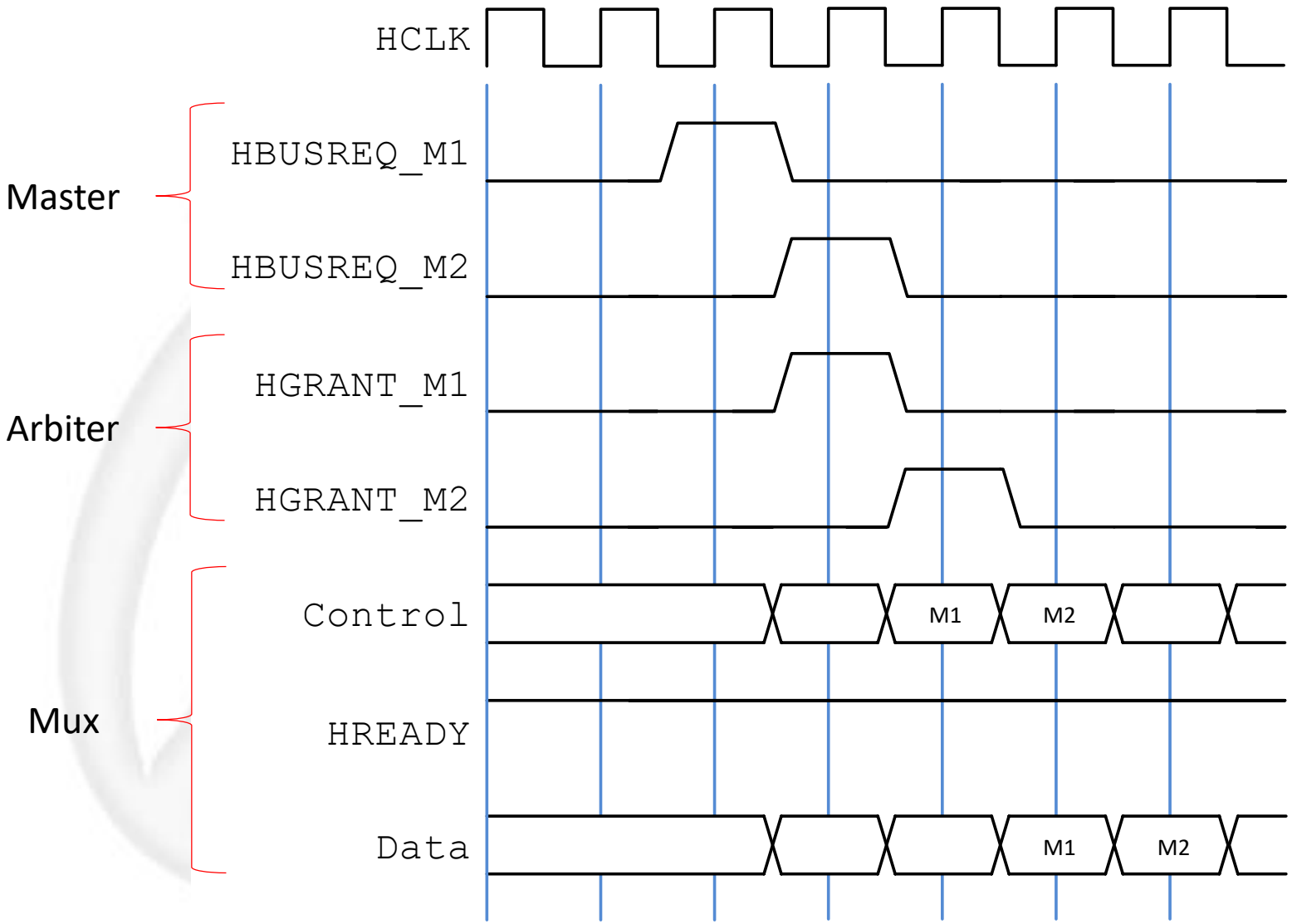
AHB Control / Response Signals



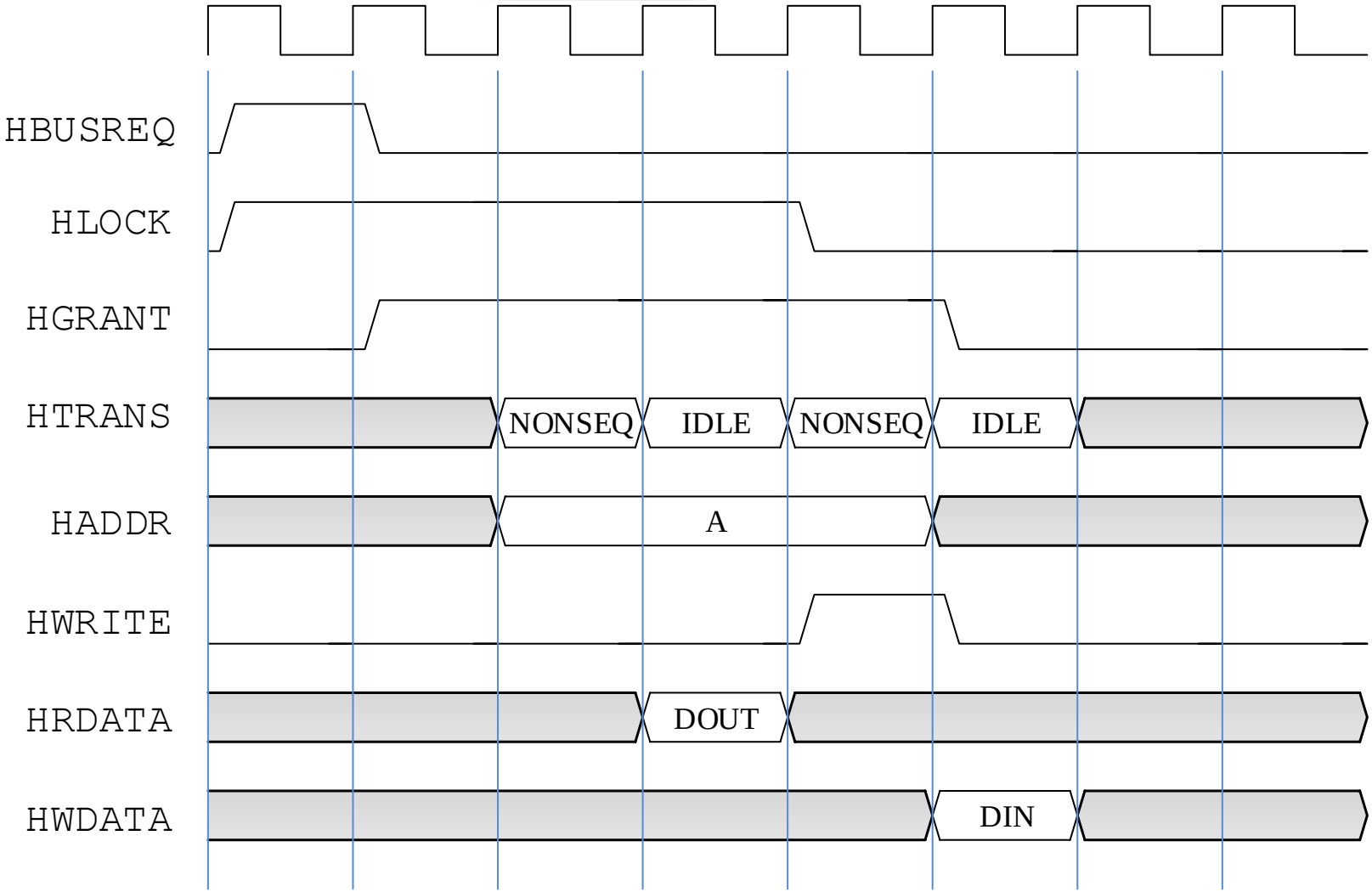
AHB Error Response



AHB Request & Grant



AHB Lock Transfer



Specification (1/2)

- Arbiter
  - Round-robin
  - Park at default master
- Decoder
  - Slave 1: 0x0000_0000 – 0x0000_ffff
  - Slave 2: 0x0001_0000 – 0x0001_ffff
  - Default slave: 0x0002_0000 – 0xffff_ffff
- Multiplexers
  - MuxM2S: Select master signals to slaves
  - MuxS2M: Select slave signals to masters

Specification (2/2)

- Default slave
  - Response ERROR when masters access (HTRANS == NONSEQ) it
- Default master
  - Always execute IDLE (HTRANS == IDLE) transfer
  - You don't need to create a default master module

Module

- Module name和module port已經定義好，請勿任意更改，Module port的意義請參考AHB的Spec
- 將AHB挖空的地方補上即可

```
always_ff@(posedge HCLK or negedge HRESETn)
begin
  if (~HRESETn)
    GrantMaster <= `AHB_MASTER_LEN'd1;
  else if (HREADY && (~NextLock))
    GrantMaster <= NextGrantMaster;
end

always_comb // define NextGrantMaster
begin
  /* complete this part by yourself */
end
```

Problem 1

作業驗證說明

SystemVerilog Assertion (1/2)

- 完成ahb.sva
  - 用assumption當成master和slave驅動AHB
  - Glue logic

```
always_comb // define address_phase
begin
  /* complete this part by yourself */
end
```

  - Property

```
// Type: HSIZE Constraint
// Ref: AMBA 2.0 Page 3-29
// Description: HSIZE should be smaller than bus width
HSIZE_data_bus_width: assume property (
  /* complete this part by yourself */
);
```

SystemVerilog Assertion (2/2)

- 不能更改ahb_monitor.svp
  - 用assert驗證AHB及驅動的時序是否正確
- 使用JasperGold驗證AHB
  - Prove結果截圖
- 報告須配合波形圖解釋你寫的Assumption
  - 波型圖可從AMBA Spec、本份說明文件、JasperGold的Visualize取得或是自繪

Verification

Table B-1: Simulation commands (Partial)

Simulation Level	Command
Problem1	
RTL	make ahb

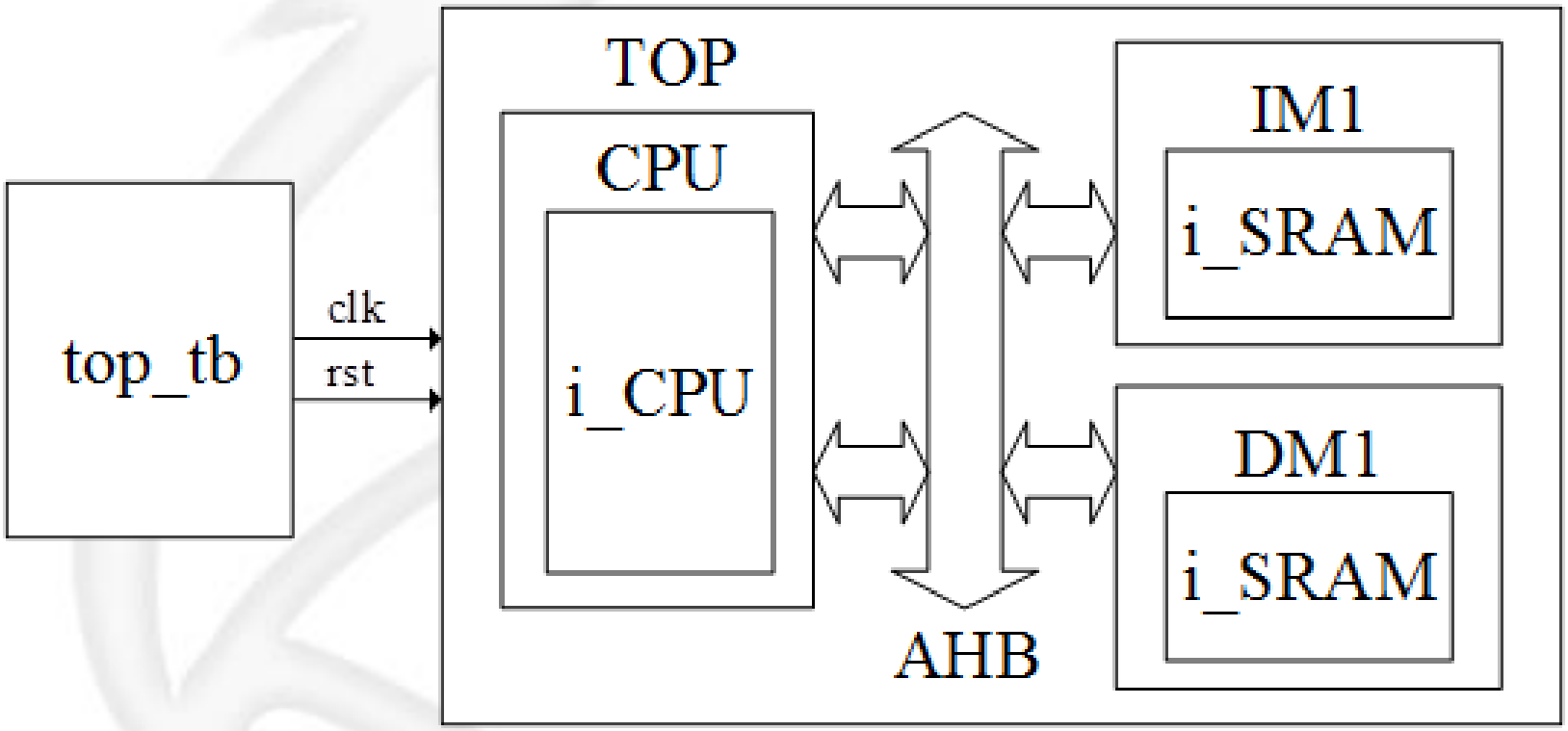
Table B-2: Makefile macros (Partial)

Situation	Command
Run JasperGold GUI with AHB verification without file pollution	make jg
Delete built files for simulation, synthesis or verification	make clean
Check correctness of your file structure	make check
Compress your homework to *tar* format	make tar

Problem 2

作業內容說明

Block Diagram



Specification

- Don’t modify any timing constraint except clock period in *DC.sdc*. Maximum clock period is 20 ns.
- Design the master wrapper between CPU and AHB.
  - Transfer Signals between CPU and AHB
- Modify SRAM_wrapper to be compatible with AHB.
- CPU has two masters
  - Instruction
  - Data
- IM (Slave 1)
  - 0x0000_0000 – 0x0000_ffff
- DM (Slave 2)
  - 0x0001_0000 – 0x0001_ffff

Module (1/2)

- Module name須符合下表要求

Category	Name			
	File	Module	Instance	SDF
RTL	top.sv	top	TOP	
Gate-Level	top_syn.v	top	TOP	top_syn.sdf
RTL	SRAM_wrapper.sv	SRAM_wrapper	IM1	
RTL	SRAM_wrapper.sv	SRAM_wrapper	DM1	
RTL	SRAM_rtl.sv	SRAM	i_SRAM	

- 紫色部分為助教已提供或已定義好，請勿任意更改
- 其餘部分需按照要求命名，以免testbench抓不到正確的名稱

Module (2/2)

- Module port須符合下表要求(同HW1)

Module	Specifications			
top	**Name**	**Signal**	**Bits**	**Function explanation**
	clk	input	1	System clock
	rst	input	1	System reset (active high)
SRAM	Memory Space			
	Memory_byte0	logic	8	Size: [16384]
	Memory_byte1	logic	8	Size: [16384]
	Memory_byte2	logic	8	Size: [16384]
	Memory_byte3	logic	8	Size: [16384]

- 紫色部分為助教已提供或已定義好，請勿任意更改
- 其餘部分需按照要求命名，以免testbench抓不到正確的名稱

Problem 2

作業驗證說明

Program

- prog0
  - 測試31個instruction (助教提供)
- prog1
  - Sort Algorithm
- prog2
  - Multiplication
- prog3
  - Division

Simulation

Table B-1: Simulation commands (Partial)

Simulation Level	Command
Problem1	
RTL	make rtl_all
Post-synthesis (optional)	make syn_all

Table B-2: Makefile macros (Partial)

Situation	Command	Example
RTL simulation for progX	make rtlX	make rtl0
Post-synthesis simulation for progX	make synX	make syn1
Dump waveform (no array)	make {rtlX,synX} FSDB=1	make rtl2 FSDB=1
Dump waveform (with array)	make {rtlX,synX} FSDB=2	make syn3 FSDB=2
Open nWave without file pollution	make nWave	
Open Superlint without file pollution	make superlint	
Open DesignVision without file pollution	make dv	
Synthesize your RTL code (You need write *synthesis.tcl* in *script* folder by yourself)	make synthesize	
Delete built files for simulation, synthesis or verification	make clean	
Check correctness of your file structure	make check	
Compress your homework to *tar* format	make tar	

VLSI Design LAB

作業繳交注意事項

Report

- 請使用附在檔案內的Submission Cover
- 請勿將code貼在.docx內
  - 請將.sv包在壓縮檔內，不可截圖於.docx中
- 需要Summary及Lessons learned
- 若兩人為一組，須寫出貢獻度
  - Ex: A(N26071234) 55%, B(N26075678) 45%
  - Total 100%
  - 自己一組則不用寫

繳交檔案 (1/2)

- 依照檔案結構壓縮成 “.tar” 格式
  - ➔ 在Homework主資料夾(N260XXXXX)使用make tar產生的tar檔即可符合要求
- 檔案結構請依照作業說明
- 請勿附上檔案結構內未要求繳交的檔案
  - ➔ 在Homework主資料夾(N260XXXXX)使用make clean即可刪除不必要的檔案
- 請務必確認繳交檔案可以在SoC實驗室的工作站下compile，且功能正常
- 無法compile將直接以0分計算
- 請勿使用generator產生code再修改
- 禁止抄襲

繳交檔案 (2/2)

- 若兩人為一組，只需一個人上傳作業到Moodle
  - 兩人都上傳會斟酌扣分
- 若兩人為一組，壓縮檔、主資料夾名稱、Report名稱、StudentID檔案內的學號都要為上傳者的學號，另一位只需在StudentID2及Submission Cover內寫上自己的學號。
  - Ex: A(N26071234)負責上傳，組員為B(N26075678)
  - N26071234.tar (壓縮檔)
    N26071234 (主資料夾)
    N26071234.docx (Report，Cover寫上兩者的學號)
    StudentID (裡面填上N26071234)
    StudentID2 (裡面填上N26075678)
- 自己一組請直接刪除StudentID2檔案

檔案結構

- 參考 Appendix A
- sim/CYCLE
  - → Specify your clock cycle time
- sim/MAX
  - → Specify max clock cycle number
- sim/prog0
  - → Don’t modify contents
- sim/progX (X ≠ 0)
  - → main.S
  - → main.c
    - Submit one of these

繳交期限

- 2018/11/01 (四) 9:00前上傳
  - 不接受遲交，請務必注意時間
  - Moodle只會留存你最後一次上傳的檔案，檔名只要是「*N26XXXXXX.tar*」即可，不需要加上版本號

VLSI Design LAB

Thanks for your participation and attendance !!