CSE 2021
COMPUTER ORGANIZATION

HUGH CHESSER
CSEB 1012U

W14-M

Agenda

Topics:

1. Memory Access
2. Review of Course Material

Patterson: 5.2, Appendix C.8, C.9

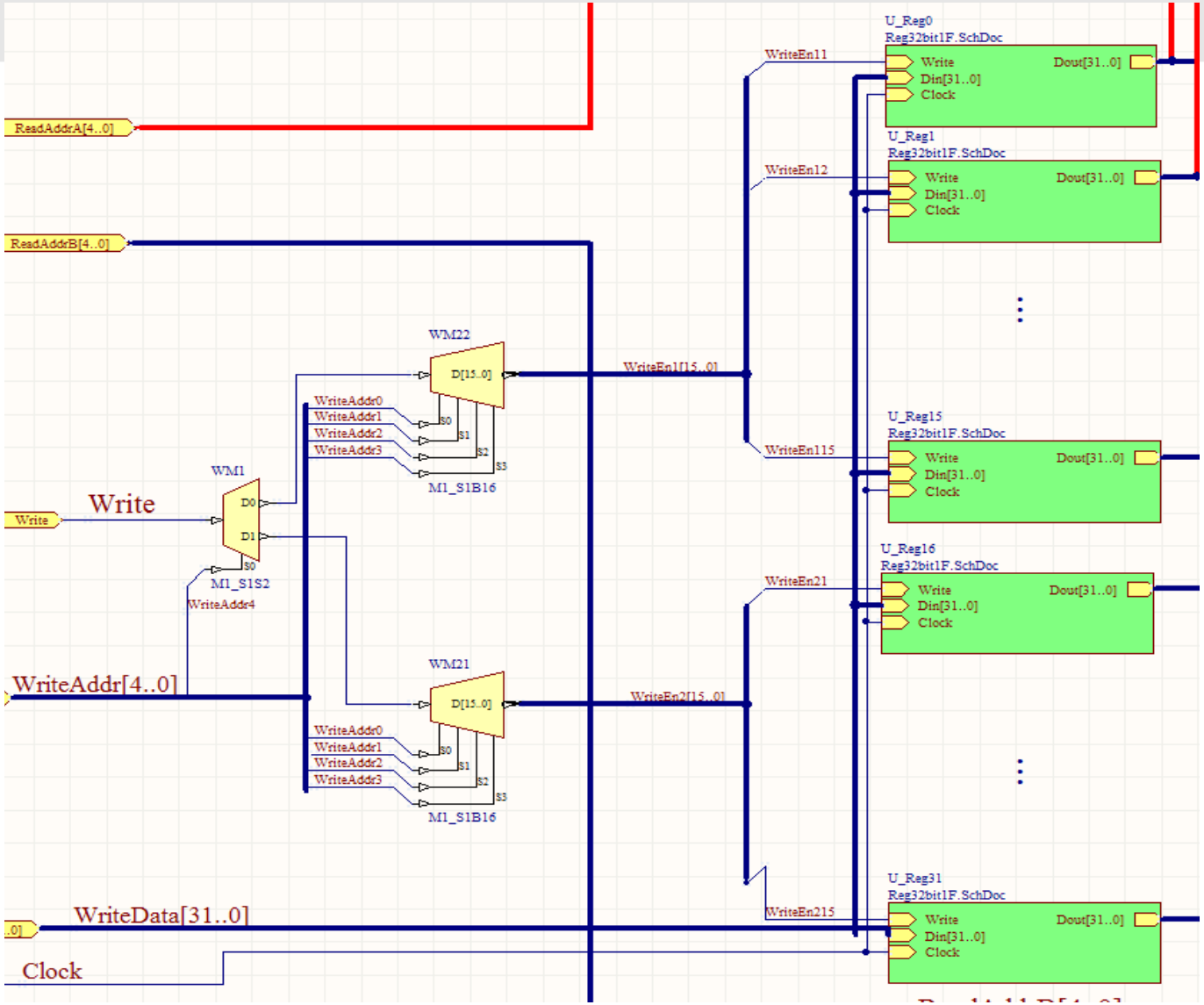
Accessing Memory

SRAM – Static Random Access Memory

- State saved in Flip-flop devices, same as the register file
- Addressing is somewhat different than decoders used for register file

Review - Register File Write

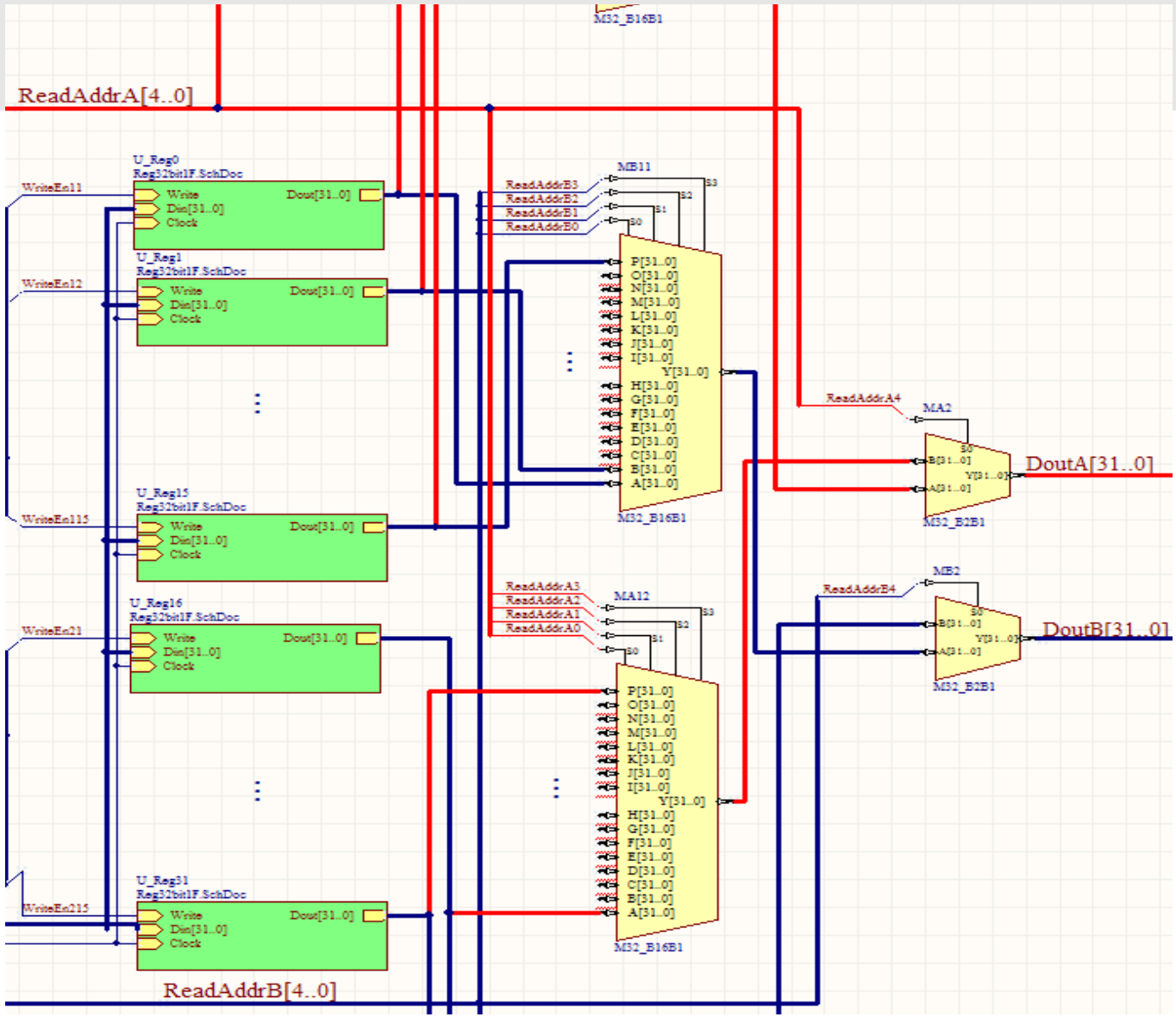
- Write Operation:
- Register number of the register to be written is one input (WriteAddr bus)
- Data to be written is the second input (WriteData bus)
- Clock that controls the write operation is the third input
- Decoders are used in the write operation



Review - Register File Read

Read Operation:

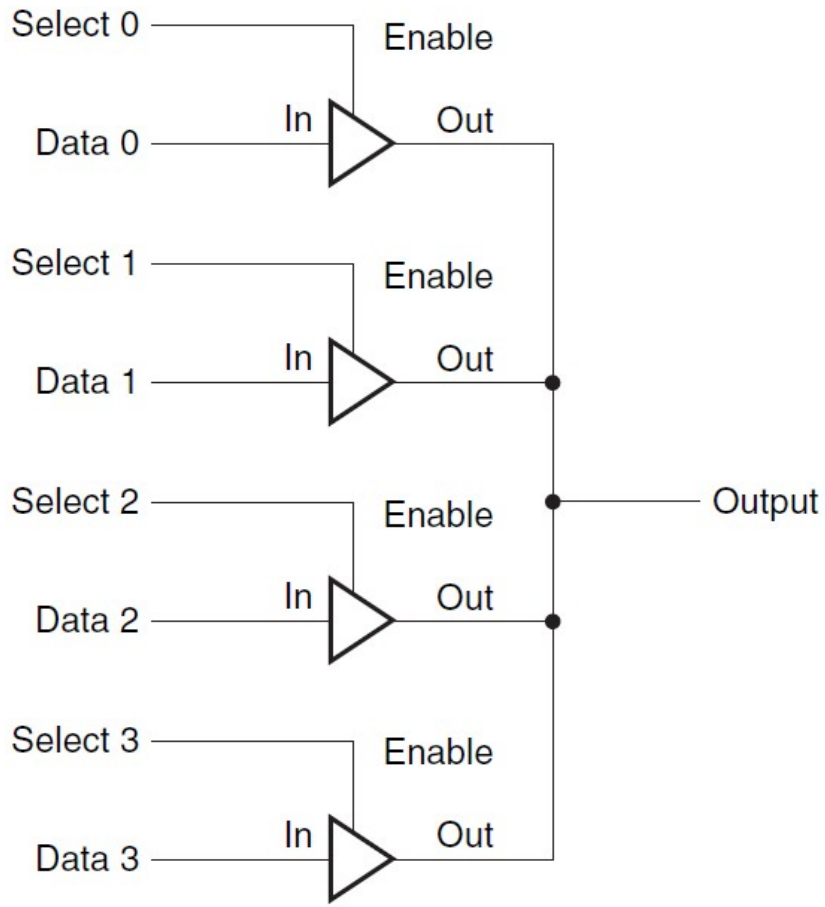
- Register number (address) of the register to be read is provided as input
- Content of the read register is the output of the register file
- Multiplexers (2 stages) are used in the read operation

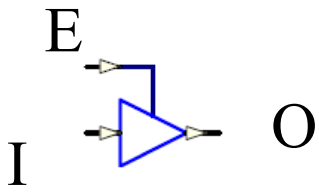


SRAM Memory Access

Read/Write Operations:

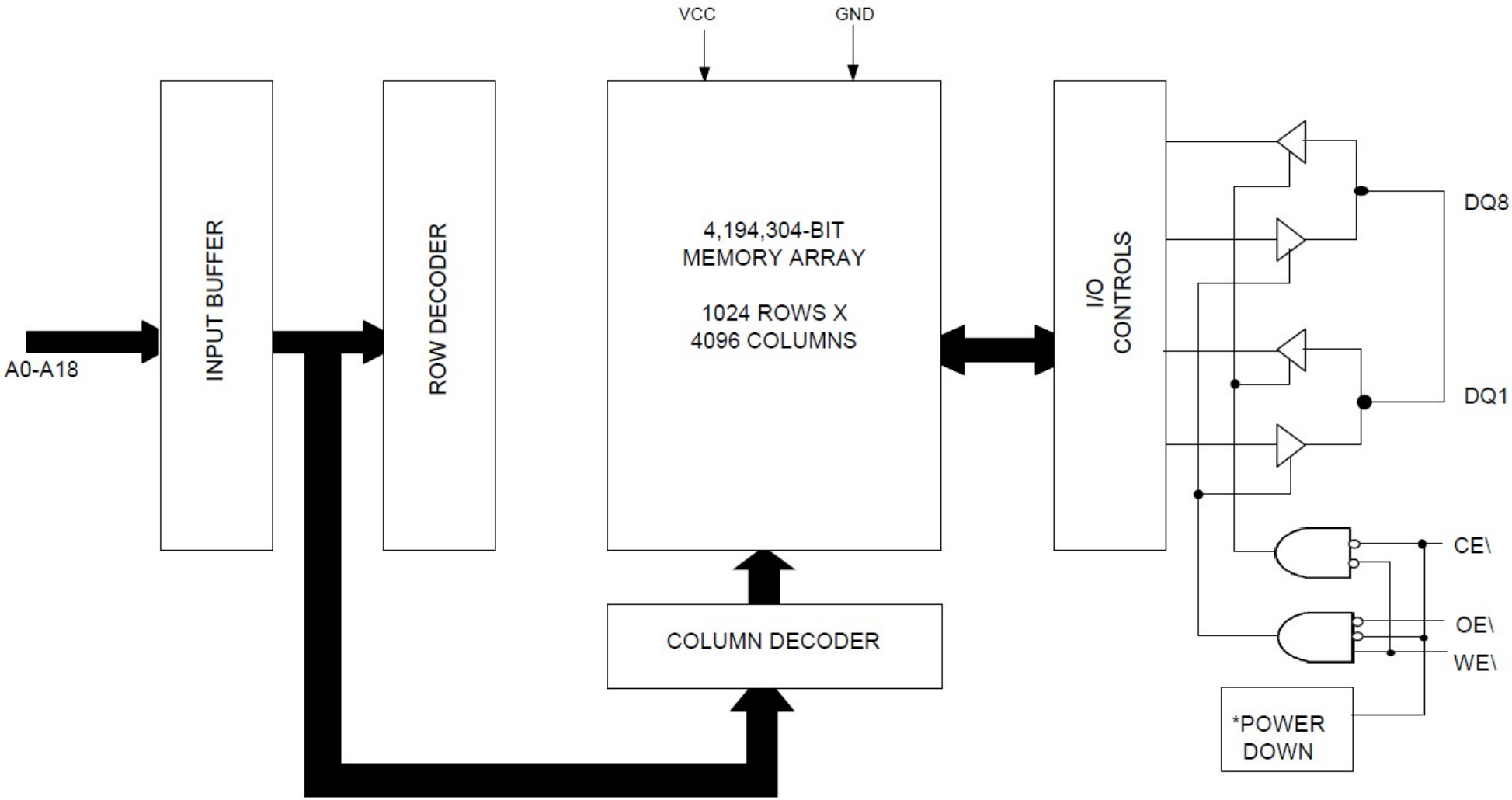
- Tri-state buffer
- Allows more that one memory cell to share the same output line



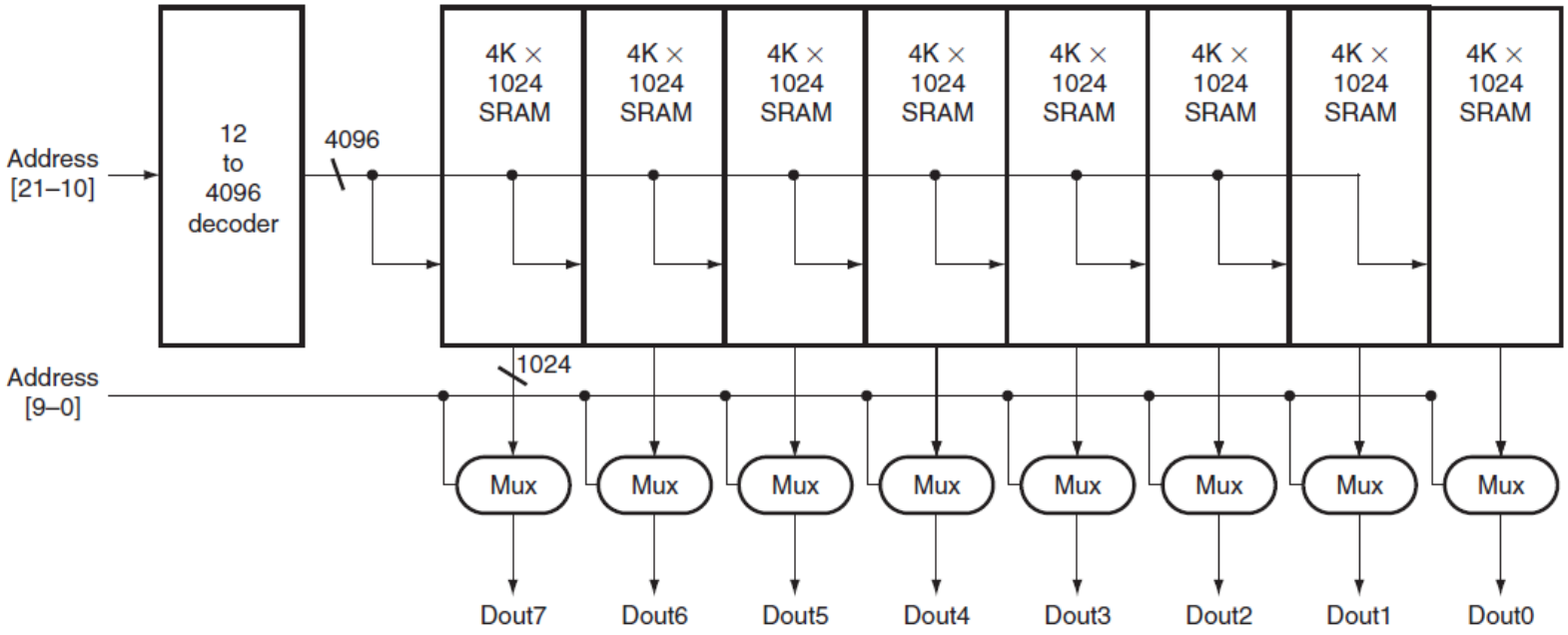


Inputs		Output
E	I	O
0	x	Z
1	1	1
1	0	0

SRAM Datasheet



Two stage Decoding Example
4M x 8 SRAM



Error Correction Codes

- Hamming codes (bit errors) or Reed-Solomon codes (multiple bits) typically used for memory checking after reads
- Based on the concept of parity – even and odd – bits are Xor'd together
- Using multiple parity bits allows bits in error to be identified and corrected

Encoding Methods

- Forward Error Correction a.k.a.
- Error Detection and Correction of Data Errors (see http://en.wikipedia.org/wiki/Error_correcting_code)

EDAC Method	EDAC Capability
Parity	Single bit error detect
Cyclic Redundancy Check (CRC)	Detects if any errors have occurred in a given structure
Hamming Code	Single bit correct, double bit detect
Reed-Solomon Code	Corrects multiple and consecutive bytes in error
Convolutional Code	Corrects isolated burst noise in a communication stream

Exam Study Suggestions

Do practice questions! Do NOT simply read the textbook

Questions are available in the back of the chapters:

- Chapters 1, 2, 3, 4, 5 (up to and including section 5.2), Appendix B, C.1 – C.10, D.3 – I have some solutions if stuck
- Practice with spim and iVerilog

Chapter 1 – Computer Terminology, Abstractions

- Instruction set architecture
- Computer performance – measures, benchmarks

Chapter 2 – Assembly Instructions

- MIPS assembly language introduction (details in Appendix B)
- Machine code
- Real, signed/unsigned number and character representations

Chapter 3 – Computer Arithmetic

- Integer addition, subtraction, multiplication, division
- Floating point – not so much

Chapter 4 – Processor Architecture

- Building blocks – logic gates, latches, flip flops
- Components – ALU, Register file, program counter, memory
- Single cycle implementation
- Multi-cycle implementation
- Pipeline implementation – control, data hazards

Chapter 5

- Memory technologies and hierarchy
- Caches

Appendices, Labs

- Appendix B – MIPS assembly and SPIM simulator details
- Appendix C – Logic design details
- Appendix D – Control – Finite state machine implementation
- Labs A – D – MIPS programming
- Labs K – N – Verilog design