Version AHG/2

EGT3
ENGINEERING TRIPOS PART IIB

Friday 7 May 2021 9 to 10.40

Module 4F14

COMPUTER SYSTEMS

*Answer not more than **two** questions.*

All questions carry the same number of marks.

*The **approximate** percentage of marks allocated to each part of a question is indicated in the right margin.*

*Write your candidate number **not** your name on the cover sheet and at the top of each answer sheet.*

STATIONERY REQUIREMENTS
Write on single-sided paper.

SPECIAL REQUIREMENTS TO BE SUPPLIED FOR THIS EXAM
CUED approved calculator allowed.
You are allowed access to the electronic version of the Engineering Data Books.

10 minutes reading time is allowed for this paper at the start of the exam.

The time taken for scanning/uploading answers is 15 minutes.

Your script is to be uploaded as a single consolidated pdf containing all answers.

1 (a) What is meant by a *load-store instruction set architecture*? In some early computer systems, a memory access would always take one clock cycle. Discuss why load-store architectures have become more attractive now that this is no longer the case. [25%]

(b) The following pseudo-code segments each compute `C = A + B`.

```
push A          load A          load r1, A
push B          add B           load r2, B
add             store C         add r3, r2, r1
pop C                           store C, r3
```

One is for a stack instruction set architecture (ISA), another for an accumulator ISA and another for a load-store ISA. Explain, with reasons, which is which. [20%]

(c) Figure 1 shows the datapath and control for a single-cycle implementation of the MIPS instructions `lw sw add sub and or slt` and `beq`. The operation times of the main functional units are: 2 ns for the memory (read or write); 2 ns for the ALU and adders; and 1 ns for the register file (read or write). All other units have negligible latencies and there is no pipelining. Consider the MIPS `addi` instruction:

Example: `addi $19,$18,-4  # $19 loaded with $18 - 4`

Format:

6 bits	5 bits	5 bits	16 bits
op	rs	rt	immediate

(I-format)

What modifications to the Fig. 1 datapath and control are required to implement the `addi` instruction? Justify your answer. [15%]

(d) Someone proposes a modification to the MIPS instruction set which removes the option of specifying an offset for loads and stores. Specifically, all load-store instructions with non-zero offsets become pseudo-instructions which are translated by the assembler into two real instructions. For example:

```
lw $15,100($2)    # reg $15 loaded with mem[$2 + 100]
```

becomes

```
addi $16,$2,100   # reg $16 loaded with $2 + 100
lw $15,$16        # reg $15 loaded with mem[$16]
```

How would you modify the Fig. 1 datapath to accommodate this change? What proportion of offset load-store instructions could be tolerated if the modification is to have a positive impact on performance? How might your answer change if the datapath were pipelined? [40%]

(cont.

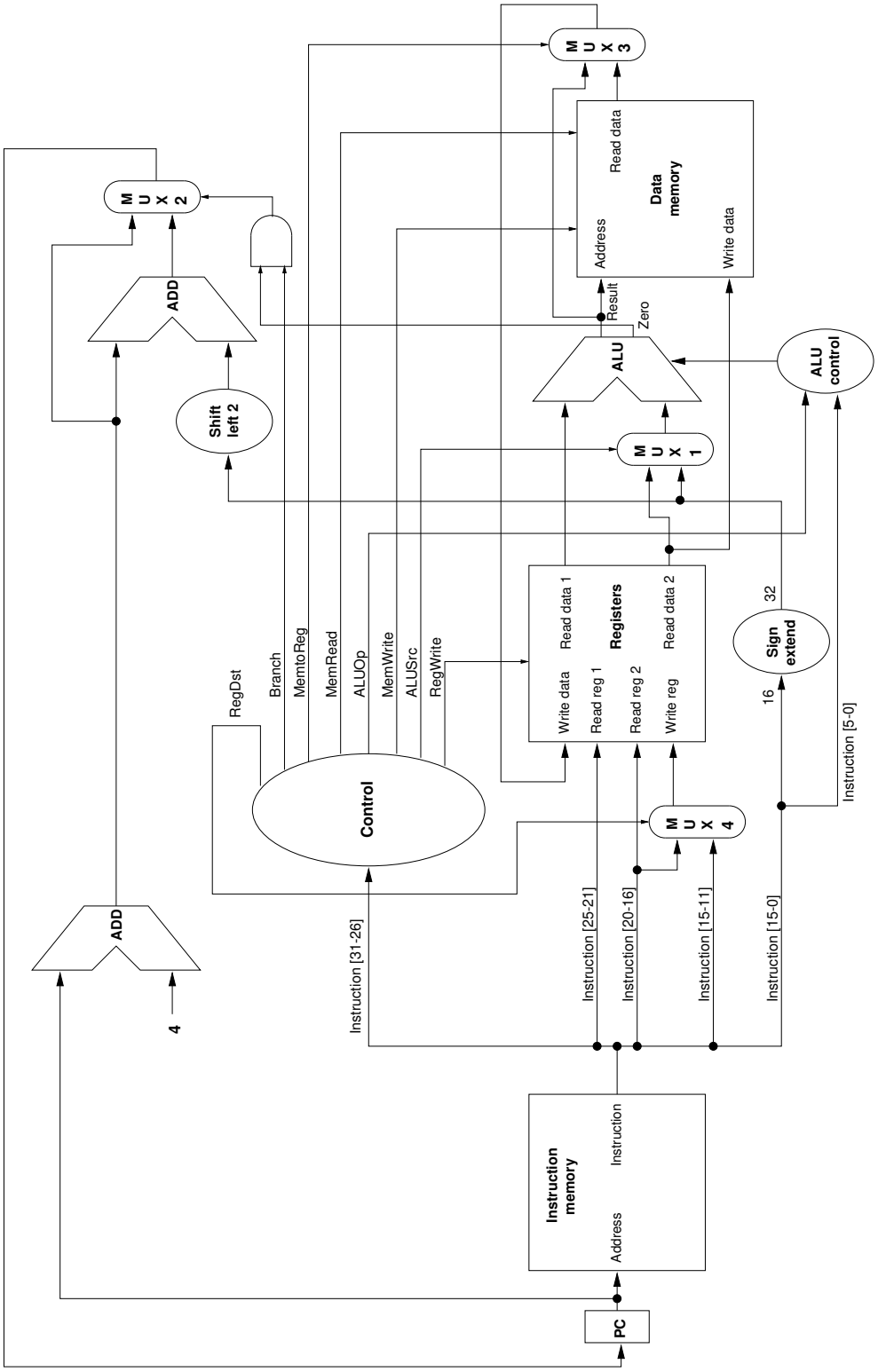

M U X 3
Data memory
Read data
Address
Write data
M U X 2
ADD
Result
Zero
ALU
ALU control
Shift left 2
M U X 1
Registers
Read data 1
Read data 2
Write data
Read reg 1
Read reg 2
Write reg
32
Sign extend
16
RegDst
Branch
MemtoReg
MemRead
ALUOp
MemWrite
ALUSrc
RegWrite
Control
M U X 4
Instruction [5-0]
ADD
4
Instruction [31-26]
Instruction [25-21]
Instruction [20-16]
Instruction [15-11]
Instruction [15-0]
Instruction memory
Address
Instruction
PC

Fig. 1

2 (a) In the context of parallel processing, define and very briefly explain the following acronyms: MIMD, SMP, UMA, NUMA. [10%]

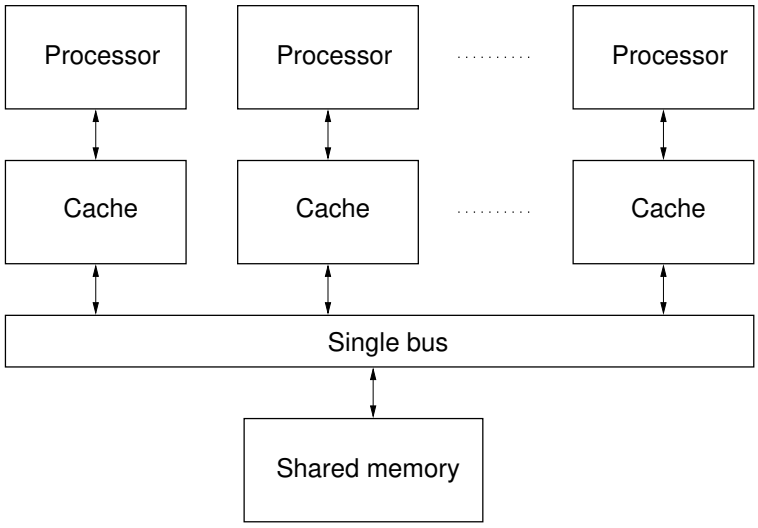
(b) Figure 2 shows the structure of two MIMD machines. Explain why smaller MIMD machines tend to be laid out like Machine A, while larger MIMD machines resemble Machine B. [10%]

(c) The individual processors in a MIMD machine can communicate through shared memory or by message passing. Is Machine B restricted to either of these alternatives? Justify your answer. [10%]

(d) Why might the optimal cache block size be smaller for a single bus SMP than for the corresponding uniprocessor? Illustrate your answer with two hypothetical memory accesses that would trigger a miss in the SMP with multi-word blocks, but not with single word blocks. [20%]

(e) A parallel processing program is run on a single bus SMP. Thread A (running on Core 1) reads a large file from disk into memory that is shared with Thread B (running on Core 2). Discuss how the hardware and operating system must cooperate to ensure that the I/O does not impose a significant load on the CPU, and that there are no undesirable side-effects the next time Thread B reads from the shared memory. [50%]

Machine A



Machine B

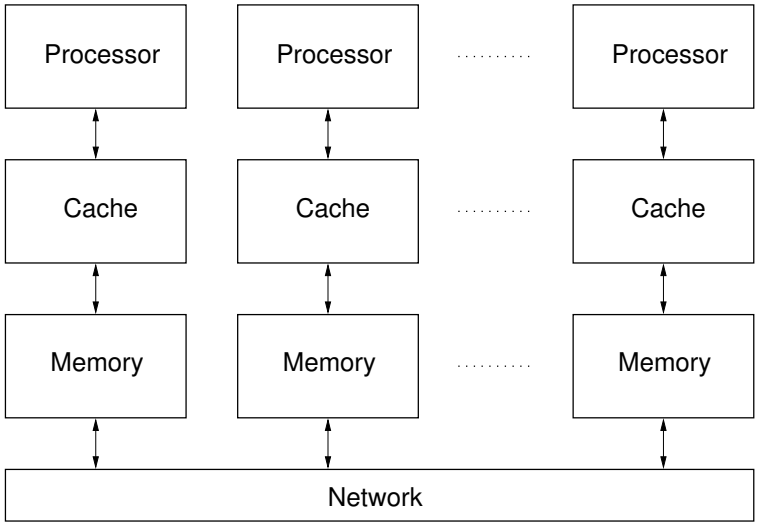


Fig. 2

3 (a) What requirements of a modern computer system motivate the adoption of a virtual memory system? [10%]

(b) Figure 3 shows the translation lookaside buffer (TLB) and one of the caches in a computer system.

(i) Is the TLB direct-mapped, set associative or fully associative? What about the cache? What is the cache block size? [10%]

(ii) Figure 3 shows the sequence of actions on a TLB hit. Describe how the data is accessed in the event of a TLB miss. [15%]

(iii) Assuming a page table contains one entry per virtual page number, estimate the memory storage requirements of a page table. How would this scale for a modern computer system with 64-bit virtual addresses? [15%]

(c) Two ways to reduce the amount of storage required for page tables are as follows.

An *inverted page table* is indexed by the physical page number. Each inverted page table entry contains the virtual page number, valid and dirty bits, and also a process identifier.

Figure 4 shows a *multi-level page table*. The top 10 bits of the virtual address are used to index the top-level *page directory*, which contains pointers to lower level page tables. The page tables are indexed by the next 42 bits of the virtual address to retrieve the physical page number. It is possible to add further levels to the hierarchy in Fig. 4.

Compare and contrast the two approaches. Your discussion should address the following points, alongside anything else you consider relevant.

- Why the inverted page table needs to store process identifiers.
- How many levels there should be in the multi-level approach.
- How rapidly each scheme can provide an address translation.
- How much storage is required by the translation tables in the two schemes.

For any quantitative comparisons, assume 64-bit virtual addresses, 8 GiB of physical memory, 4 KiB pages and 16-bit process identifiers. [50%]

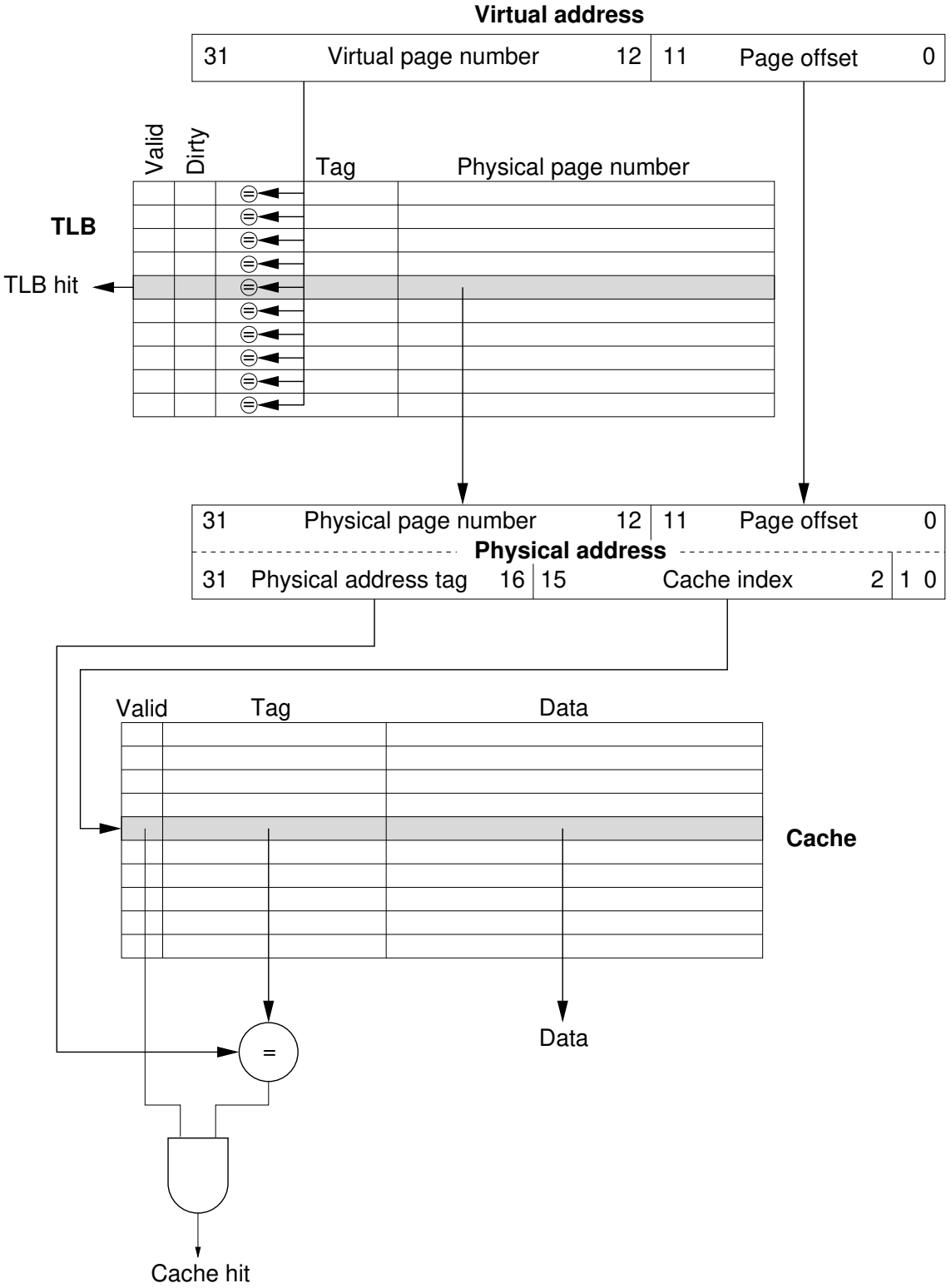

Virtual address
31
Virtual page number
12
11
Page offset
0
Valid
Dirty
Tag
Physical page number
TLB
TLB hit
31
Physical page number
12
11
Page offset
0
Physical address
31
Physical address tag
16
15
Cache index
2
1
0
Valid
Tag
Data
Cache
Data
=
Cache hit

Fig. 3

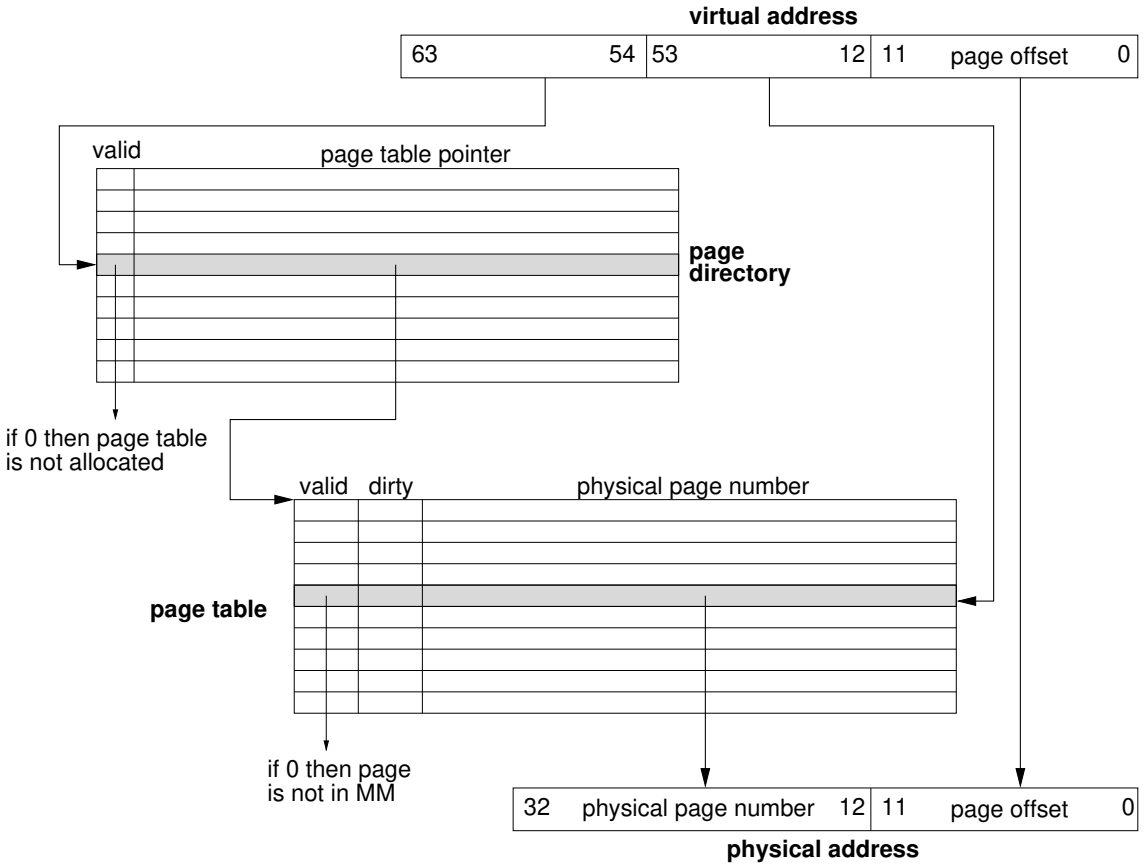


Fig. 4

END OF PAPER

Part IIB 2021

Module 4F14: Computer Systems

Numerical Answers

1. (d) 33% offset load-store instructions could be tolerated without pipelining

3. (b) (i) TLB is fully associative, cache is direct-mapped, cache block size is one word
   (iii) ~ 3 MiB for the 32-bit system