Version AHG/2

EGT3
ENGINEERING TRIPOS PART IIB

Thursday 26 April 2018 2 to 3.40

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.*

STATIONERY REQUIREMENTS
Single-sided script paper

SPECIAL REQUIREMENTS TO BE SUPPLIED FOR THIS EXAM
CUED approved calculator allowed
Engineering Data Book

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

You may not start to read the questions printed on the subsequent pages of this question paper until instructed to do so.

1 (a) Consider the pipelined MIPS datapath in Fig. 1. The execution time of the ID, EX and WB stages is 1 ns, whereas the IF and MEM stages require 2 ns. Calculate the maximum asymptotic speed-up compared with the non-pipelined version of the datapath. Why is this speed-up unlikely to be achieved in practice? [20%]

(b) A data forwarding unit is to be added to the MIPS datapath in Fig. 1. Sketch the relevant part of the enhanced datapath, showing how the forwarding unit is connected to the other components. [20%]

(c) The following extract of MIPS code increments all the elements of an n-element array by the contents of `$10`. Each element of the array is four bytes. The starting address of the array is `Astart` and $4n$ is in `$11`.

```
       add $9,$0,$0       # clear $9 to zero
Loop:  lw $8,Astart($9)   # $8 loaded with data at address $9+Astart
       add $8,$8,$10      # $8 loaded with $8+$10
       sw $8,Astart($9)   # $8 stored at address $9+Astart
       addi $9,$9,4       # $9 loaded with $9+4
       bne $9,$11,Loop    # Jump back 4 instructions if $9≠$11
```

The code is run on a 2-way superscalar version of the enhanced datapath in (b), with no hardware restrictions on instruction sequencing. Discuss how a compiler might optimize the MIPS code to take full advantage of the superscalar resources. Consider re-ordering instructions and/or unrolling the loop so that multiple array elements are incremented each time around. [40%]

(d) The optimizations considered in (c) are known as *static pipeline scheduling*, since they are determined at compile-time by the compiler. The alternative is *dynamic pipeline scheduling*, where the hardware rearranges the instructions at run-time. Discuss the relative advantages and disadvantages of the two approaches. [20%]

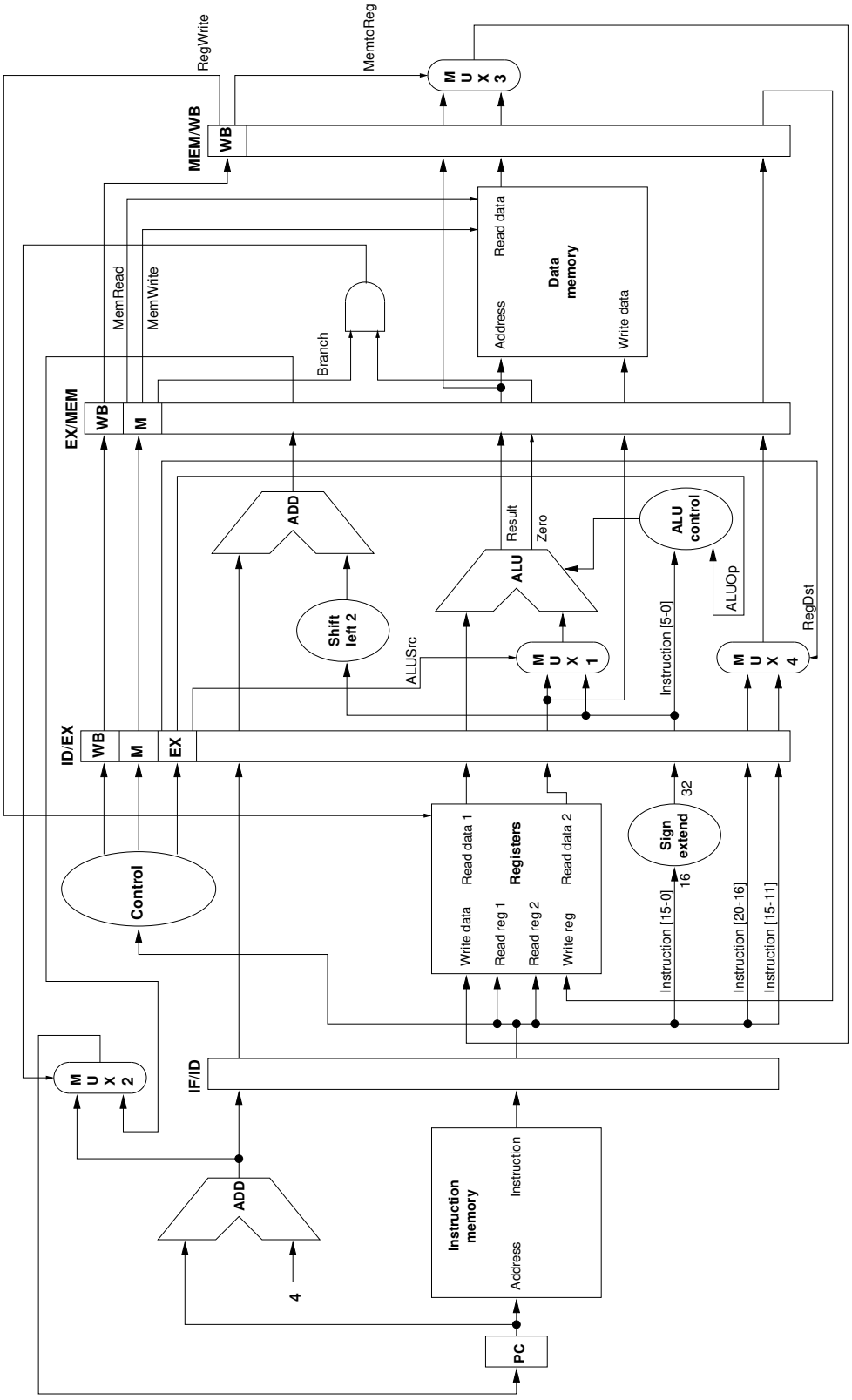
RegWrite
MemtoReg
M U X 3
MEM/WB
WB
Data memory
Read data
Address
Write data
MemRead
MemWrite
Branch
EX/MEM
WB
M
ADD
Result
Zero
ALU control
ALU
ALUOp
Shift left 2
ALUSrc
M U X 1
Instruction [5-0]
M U X 4
RegDst
ID/EX
WB
M
EX
32
Sign extend
16
Control
Registers
Read data 1
Read data 2
Write data
Read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

Fig. 1

2 (a) What requirements of a modern computer system motivate the adoption of a virtual memory system? [15%]

(b) A process accesses the following sequence of virtual pages (given by their virtual page numbers): 1 0 2 3 0 1. Assuming that the process is allocated three physical pages, and the page replacement strategy is LRU, how many page faults are there? [15%]

(c) Consider a computer system with a 32-bit virtual address and 4 KB pages. If each row of the page table occupies 4 bytes, how much memory is required to store a single-level, linear page table? How many such page tables are required? [15%]

(d) Figure 2 shows an alternative, 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 10 bits of the virtual address to retrieve the physical page number. Assuming each row of each table occupies 4 bytes, how much memory is required to store the page directory and page tables of a process that accesses virtual page numbers in the range 0–1023 and 1047552–1048575? [20%]

(e) Now consider a computer system with a 64-bit virtual address and 4 KB pages. It is proposed to add further levels to the hierarchy in Fig. 2, such that each table fits in one page of memory. How many levels are required? Why are one-page tables a good idea? [20%]

(f) Discuss briefly how the computational expense of virtual-to-physical address translation depends on the number of levels in the hierarchy. [15%]

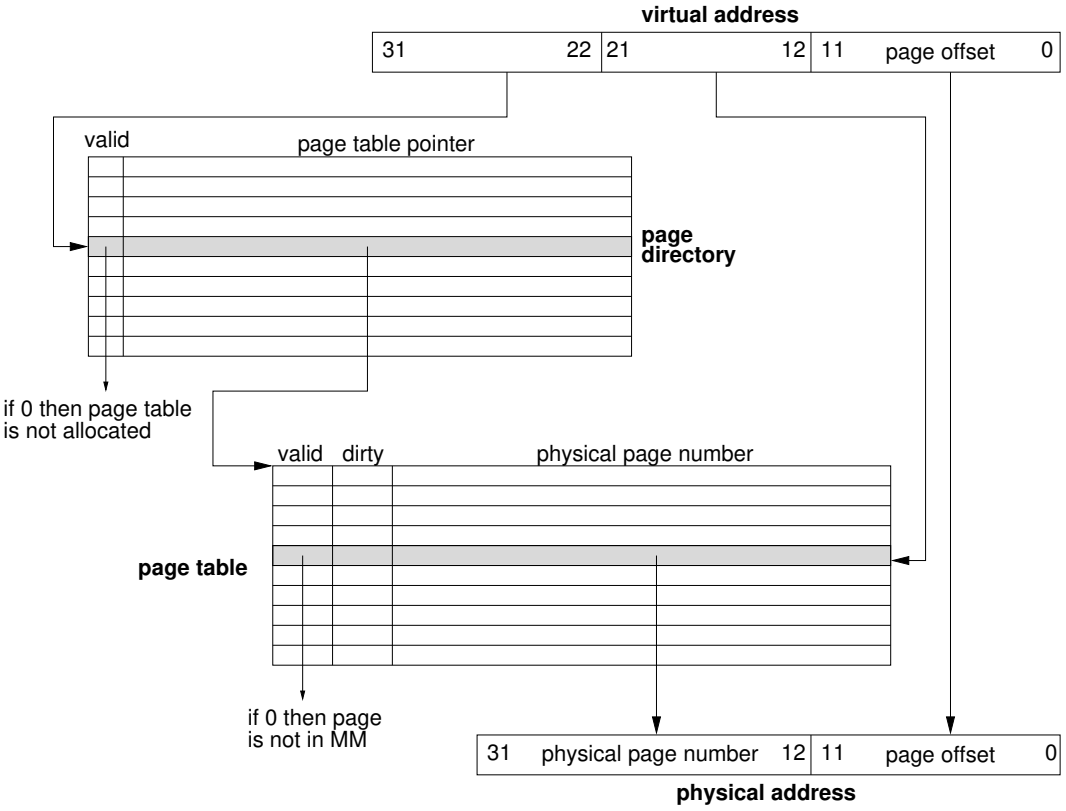
virtual address
31
22
21
12
11
page offset
0
valid
page table pointer
page
directory
if 0 then page table
is not allocated
valid
dirty
physical page number
page table
if 0 then page
is not in MM
31
physical page number
12
11
page offset
0
physical address

Fig. 2

3 (a) In the context of parallel processing, define and very briefly explain the following acronyms: SIMD, MIMD, SMP, UMA, NUMA, SMT. [30%]

(b) Figure 3 shows a state transition diagram for a single-bus SMP's bus-snooping cache coherency protocol. The diagram relates to a particular block (location) in the cache. Stimuli from the local CPU are shown in italics, with actions shown in upright text. Thus, for example: a CPU read miss for a block in the "modified" state occurs when the CPU reads a block that is not cached and needs to reside at this location; this triggers a write-back of the current block and a broadcast of the miss on the bus; the new block is subsequently provided by the shared memory or another cache; the newly cached block is now marked as "unmodified", since it has not been written to by the local CPU.

(i) Figure 3 is incomplete in that it shows only transitions and actions triggered by reads and writes from the *local* CPU. It does not show transitions and actions triggered by broadcasts from *other* CPUs on the bus. Draw a state transition diagram, with the same three states, showing the possible stimuli from *other* CPUs, and the corresponding transitions and actions. [45%]

(ii) CPUs A and B are part of the single-bus SMP. Copy Fig. 4 and place in each box a tick or a cross, indicating the allowable states of corresponding cache blocks. [25%]

(cont.

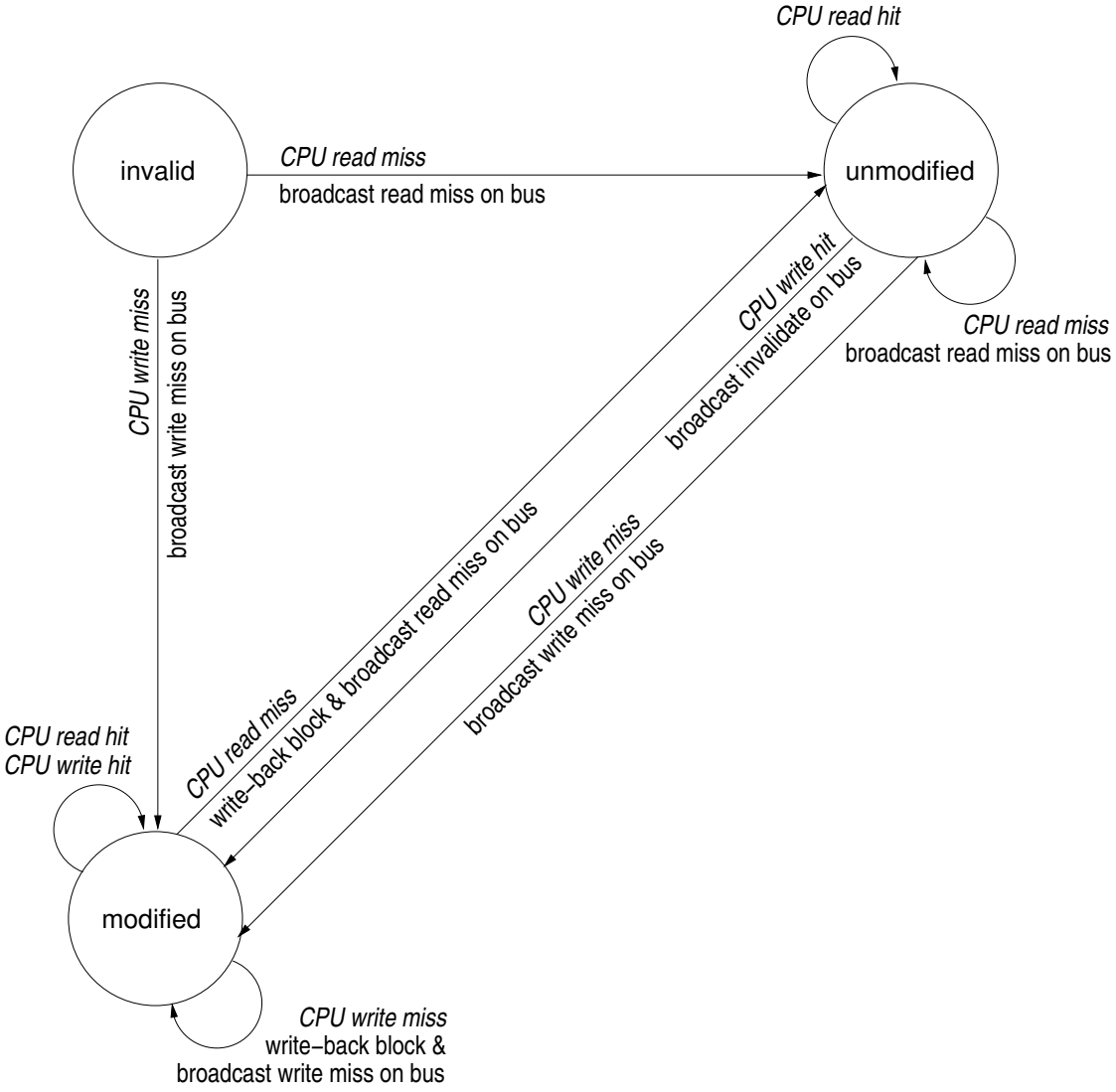

Fig. 3

CPU A \ CPU B	invalid	unmodified	modified
invalid			
unmodified			
modified			

Fig. 4

END OF PAPER

THIS PAGE IS BLANK

Part IIB 2018

Module 4F14: Computer Systems

Numerical Answers

1. (a) $\times 3.5$

2. (b) 5
   (c) 4 MB
   (d) 12 KB
   (e) 6