PAPER C112

INTRODUCTION TO COMPUTER SYSTEMS

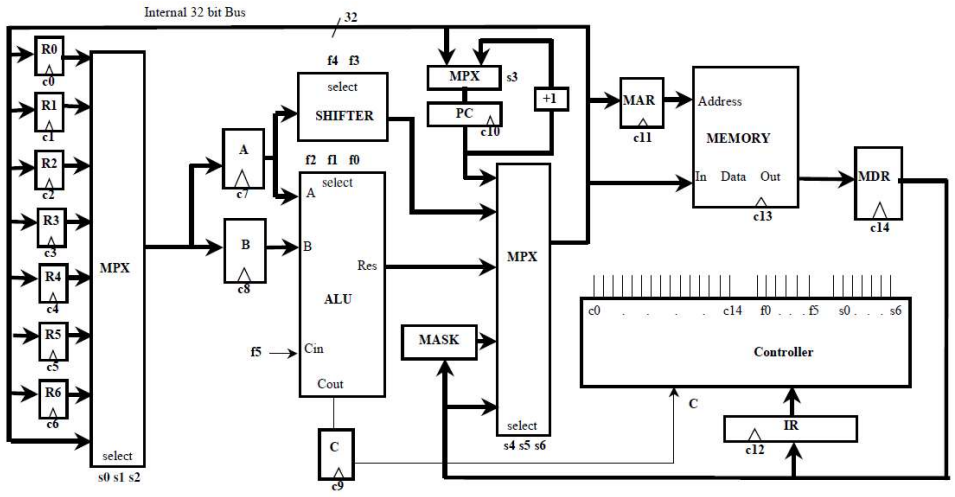
Friday 15 May 2020, 11:00
Duration: 80 minutes
Post-processing time: 30 minutes
Answer TWO questions

While this time-limited remote assessment has not been designed to be open book, in the present circumstances it is being run as an open-book examination. We have worked hard to create exams that assesses synthesis of knowledge rather than factual recall. Thus, access to the internet, notes or other sources of factual information in the time provided will not be helpful and may well limit your time to successfully synthesise the answers required.
Where individual questions rely more on factual recall and may therefore be less discriminatory in an open book context, we may compare the performance on these questions to similar style questions in previous years and we may scale or ignore the marks associated with such questions or parts of the questions. In all examinations we will analyse exam performance against previous performance and against data from previous years and use an evidence-based approach to maintain a fair and robust examination. As with all exams, the best strategy is to read the question carefully and answer as fully as possible, taking account of the time and number of marks available.

Paper contains 2 questions

1 A Manual Processor with Memory

A 32-bit manual processor with memory has the following block diagram:

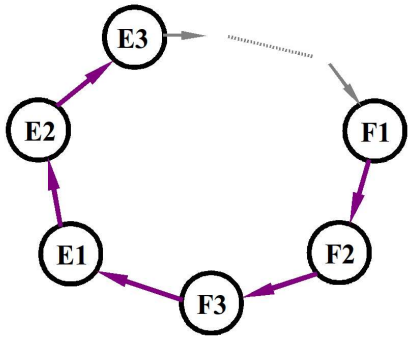


The function of the ALU is defined by the following table:

Selection Bits	000	001	010	011	100	101	110	111
Function	0	B-A	A-B	A plus B	A xor B	A+B	A·B	-1

The function of the shifter (by selection bits) is (00) hold, (01) shift left, (10) shift right arithmetic (duplicating the top bit), and (11) rotate right (setting the top bit to bit zero).

The controller is a finite state machine that first fetches a program instruction from the memory and then provides the correct operation sequences to execute that particular instruction. Hence, there are some fetch states (F) followed by some execute states (E):



a How could we realise a one bit memory? In order to have more than one bit of memory available, what would we need to be able to address several one bit memories? Is a static Random Access Memory (RAM) a combinational or sequential circuit? Explain. How does this hold for dynamic RAM? Explain.

b You must now define a part of the instruction set. The top eight bits (bits 31-24, if the lowest bit is bit 0) of the 32 bit instruction word will define the instruction ("opcode"). The next four bits (23-20) store the index of a destination register (Rdest). The data (such as an address) carried in the instructions (if any) are stored in bits 19-0. This enables us to separate out any data from the opcode with a very simple MASK circuit shown on the data path diagram in the first figure. Draw the circuit of the MASK. Does it need gates? What is the disadvantage of this design choice?

c The memory direct reference instructions are now to be defined. In the following table, there is an example given for LOAD. Complete for STORE, JUMP, and CALL.

Instruction	Cycle	Transfers	Path
LOAD Rdest, Address	E1 E2 E3	MAR←MDR MDR←Memory Rdest←MDR	Use MASK No MASK
STORE Rdest, Address	E1 E2		
JUMP Address	E1		
CALL Rdest, Address	E1 E2 E3		

d In order to allow for usage of more bits when addressing the memory, we next introduce indirect reference instructions. In these, we use two register instructions, with an additional source register (Rsrc) defined in the bits (19-16), hence, leaving the bits (15-0) as unused. Now, the memory indirect reference instructions are to be defined. In the following table, there is an example given for LOADINDIRECT. Complete for STOREINDIRECT, JUMPINDIRECT, and CALLINDIRECT.

Instruction	Cycle	Transfers	Path
LOADINDIRCET Rdest, Rsrc	E1 E2 E3 E4	A←Rsrc MAR←A MDR←Memory Rdest←MDR	 No MASK
STOREINDIRECT Rdest, Rsrc	E1 E2 E3		
JUMPINDIRECT Rsrc	E1 E2		
CALLINDIRECT Rdest, Rsrc	E1 E2 E3		

e How could you make the processor go faster? Name four suggestions.

The five parts carry equal marks.

Section B (Use a separate answer book for this Section)

2 Boolean algebra, Combinational Circuit Design, and Number representation

a Using Boolean algebra, simplify the following Boolean expression to its simplest form (fewest number of literals):

$$E = A' \cdot (A + B) + (B + A \cdot A) \cdot (A + B')$$

where $\cdot$, $+$, and $'$ represent "AND", "OR" and "NOT" operations respectively. Please show the sequence of your steps and state the reduction rules used explicitly.

b The full adder is defined in terms of its inputs A, B and C_{in} as

$$C_{out} = A \cdot B + C_{in} \cdot (A \oplus B)$$
$$S = A \oplus B \oplus C_{in}$$

Express both Boolean equations in

i) the canonical minterm form.

ii) the canonical maxterm form.

c Design a four-bit two's complement subtractor, with a carry input and a carry output, using four full adders and some inverter gates. (Note that the the carry input is provided to allow high precision integer arithmetic to be performed, and will be set to 1 when the least significant bits of the binary numbers are being processed.)

d Design a simple four bit ALU circuit which has a single one-bit function select input F, two four-bit inputs A and B, a one-bit carry input (Cin) and a one-bit carry output ($Cout$). If $F = 0$ the output of the ALU is A+B+Cin, if $F = 1$ the output of the ALU is A-B-Cin. You can use the circuits you designed in parts a and b, a four-way two-input multiplexer and a single-way two-input multiplexer as functional blocks in your circuit.

e Express the following decimal number **-139.325** in binary and hexadecimal using the IEEE Single Precision Format. Show your working clearly. If you use extra bits in your calculation, use rounding rather than truncation when calculating the result.

The five parts carry equal marks.