COMP12111 Exam Feedback 2019-20

Q1. Multiple Choice - CM

83% got this correct

Q2. Multiple Choice - PWN

Easy one this. The most popular incorrect answer was Hertz, which is the units if frequency.

93% got this correct

Q3. Multiple Choice - CM

90% got this correct

Q4. Multiple Choice - PWN

The processor design has a 12-bit address bus, which means it has the capability of addressing 2^{12}, or 4,096 memory locations. The most popular incorrect answer was 256 – which is 2^{8}, but the size of the data bus does not tell us how many locations are available, just the size of each location.

89% got this correct

Q5. Multiple Choice - PWN

You were asked to identify the incorrect statement, which is that “The address bus is 16 bits wide”, which is incorrect because the MU0 address but is 12 bits. The most popular incorrect answer was that “MU0 arithmetic instructions operate on one operand from memory”, which is true as the instruction provides an address to an operand – the other operand comes from the accumulator.

73% got this correct

Q6. Multiple Choice - CM

77% got this correct

Q7. Multiple Choice - PWN

You were required to identify how long it takes to execute an instruction if the clock frequency is 50MHz. A clock frequency of 50MHz has a clock period is 20ns. As each instruction takes 2 clock cycles to execute, so the total time for each instruction is 40ns. The most popular incorrect answer was 20ns – you fell for my trick!

32% got this correct

Q8. Multiple Choice - CM

65% got this correct

Q9. Multiple Choice - CM

96% got this correct

Q10. Multiple Choice - PWN

You were required to identify the action of the JNE instruction, which is to branch to the specified address in memory if the result of the previous instruction is not equal to zero. The most popular incorrect answer was that the last instruction executed produced a negative result, which is JGE! Easy question overall.

93% got this correct

Q11. Multiple Choice - PWN

This question takes some thought as it involves determining the critical path knowing the actions that take place in the fetch and execute cycles. In fetch we do a memory access and PC increment at the same time, so the maximum delay is 6ns. In the execute, the maximum delay takes place for an arithmetic operation where we need to read an operand from memory (takes 6ns) and perform the operation (takes 4ns), so the maximum delay is 10ns, which sets the minimum period of the clock. So, the maximum clock frequency is 1/10ns = 100MHz, so choose 100MHz. The most popular incorrect answer was 70MHz.

36% got this correct

Q12. Multiple Choice - PWN

You were to identify the correct answer, which is that "Handshaking is a technique that allows an I/O device to acknowledge when it has received sent data". The other statement were incorrect: as strobing is an asynchronous technique, the use of interrupts frees up processor time, and the programmer has control when software interrupts occur. The most popular incorrect answer was "Strobing is a synchronous technique for data transfer between CPU and I/O".

78% got this correct

Q13. Multiple Choice - PWN

In order of increasing capacity, the memory hierarchy is ordered as follows: register, main memory, cache, flash, hard disk, DVD. So the correct answer from those given is Register, Flash, Hard Disk. The most popular incorrect answer was “Cache, Flash, Main Memory”. Straightforward question.

90% got this correct

Q14. - CM

This question aimed to test basic knowledge of half-adders/full adders.
Issues:

- Caution was needed on how the half adders are connected to produce the full adder (and to explain how they are connected clearly).
  The question asked for the expressions of the half-adder: S = (~A.B) + (A.~B) i.e. A XOR B, Co = A.B i.e. A AND B, not for some general Verilog code (e.g. assign {Co, S} = A + B + cin;). However, if the formulas were shown in Verilog code, this was taken as correct.
- The question explicitly asked how to use the half adder to build the 1-bit full adder, not to calculate the final full adder sum and cout formulas. However, whenever this goes done thoroughly and derived correctly, it got full marks.

Average 1.55/3 (52%)

Q15. - CM

The aim here was to demonstrate a basic understanding of the critical path; the maximum delay through the adder, which is data dependent. The main reason that this type of adders are slow is that the carry needs to propagate through every bit in the adder. The approximate delay for an N-bit adder, as instructed during the lectures, will be ~ delay x N = 396 ns.
Issues:

- Caution was needed in the formulation of the question. The delay of the full 1-bit adder is given. So, there is no need to worry about the number of gate delays and calculating worst-case propagation delay path inside each adder. Consequently, the aim was not to use a formula calculating number of gate delays since the total max delay of the 1-bit full-adder is given as an approximate value. If you assume that there is a carry propagation throughout the N-bit adder (it is data-dependent) then the approximate full delay would be ~ delay x N = 396 ns in this case.
- Another point is that it's not certain that we can assume the gates have equal delay. **However**, if the formula was correctly used to calculate the delay of each gate and applied to calculate the total delay (with a longer delay for the first adder, from input to carry: 3 gate delays, and subsequently 10*2, for carry propagation in later adders: 2 gate delays) in order to have (2n+1)*delay=(2*11+1)*12=276ns, this was also taken as correct.
-

Average 1.95/3 (65%)

Q16. - CM

Most answers here were correct.

Issues:

- They are referred to as universal gates because combinations of them can be used to accomplish any of the basic operations.
- NAND was more often incorrect / partially right. The correct answer is that a NAND gate is True only if **any** or **both** of its inputs A and B are false.

Average 3.26/4 (82%)

Q17. - CM

The goal here was to emphasise the need for CAD tools for allowing/helping the designer to manage the complexity of modern processors and enable today's leading-edge computing technology.

Issues:

- The question did not ask for a description of what CAD tools can do (or a description of the waveform viewer), but what is the need they fulfill (see the goal above).
- Designing and testing before building hardware, was not enough to describe the target of this question.
- Describing the concepts hierarchy and abstraction by themselves was not the target of this question.
- Managing today's chips complexity and enabling design of leading-edge devices is the two key things that an answer needed to have.

Average 1.37/2 (69%)

Q18. - CM

The goal here was to demonstrate knowledge the four basic **values** a signal can assume, as can be seen, e.g. in timing diagrams.

Issues:

- This is about the **values** the signals can assume (low (0), high (1), unknown (X), tristate/high-impedance (Z)), not the signals that will be shown. So, answers talking about clock signal, inputs, output signal, reset signal, CE signal, etc, were all *not* correct. A big proportion of the answers included such signals, rather than the correct answer.
- Unknown (X) value is neither true or false. Some people have called this "error (X)".
- Undefined / "don't care" is not two different states. Some answers got this wrong and missed the opportunity to describe the tristate (Z) / high-impedance/disconnected state.

- Writing only high or low, without 1 or 0 or other explanation is taken as fully correct. Stating only X or Z (or some other letter), without a description, was not accepted as a correct answer.

Average 1.07/2 (54%)

Q19. - CM

This question aimed to demonstrated basic understanding of Verilog and systems of counters. A range of answers was provided.
Issues:

- Only one always block was enough but other solutions (*Any variation*) that worked were accepted as correct, e.g. 3 always modules solution, similar to what we did in the labs, was also accepted as correct (if it had the correct functionality).
- The reset, according to the question's specifications needed to be synchronous. This was often mixed up with asynchronous.
- Use of blocking and non-blocking was often mixed up. In particular using blocking with always@(posedge clock) instead of non-blocking, was a mistake that appeared frequently.
- Use of sensible commenting got a bonus point.

Average 4.54/6 (76%)

Q20. - PWN

This question tests your understanding of the 3-box model and the system bus. Marks were awarded as follows:

- identifying the three elements of the 3-box model: CPU, memory, I/O – 1 mark
- identifying that the system bus contains the address bus, data bus and control signals – 1 mark

Issues:

- not answering all the question
- discussing that the systems bus connects the three components and not being clear what is in the systems bus.
- the system bus is more than a group of wires.
- missing out the control bus
- memory control – although I didn't penalise for this – the control bus contains other signals, such as the interrupt.
- restrict the discussion to what is asked in the question!

On the whole the question was answered very well.

Average 1.65/2 (83%)

Q21. - PWN

This question tests your understanding of the operation of the DRAM cell, requiring you to discuss the operation when writing data and reading data. I wanted the following points highlighted in the discussion:

a) write operation

- word line is taken high
- switches on the transistor (I accept connects the capacitor to the bit line)
- apply data to be stored to the bit line
- if storing a ‘1’ then the capacitor is charged

b) read operation

- word line is taken high
- switches on the transistor to connect the capacitor to the bit line
- sense the data stored in the capacitor
- as it discharges

Issues:

- Not describing the writing and reading process and instead just focusing on the general operation.
- Stating that the data comes from the word line, whereas the word line turns on the cell for reading or writing.
- general clarity of the answers provided – not enough detail!
- general misconceptions of the operation.

On the whole the question was poorly answered. Only a handful of students provided good, clear answers that achieved full marks.

Average 1.95/6 (33%)

Q22. - PWN

This question tests your understanding of the one-dimensional addressing scheme, it’s operation and to identify a disadvantage.

Marks were awarded for raising the following points with regards to the operation:

- the address is decoded by a decoder (here a 4-to-16 line decoder)
- results in only one of 16 lines going high, the other remain low
- to enable/select the memory location being used (read/write action)

Possible disadvantages (just one required for the one mark):

- does not scale well for increased memory sizes

- the decoder can become overly complex

Issues:

- not being clear that only one location in memory is enabled, i.e. only one signal from 16 goes high, the other remain low
- lack of clarity in the answer
- complexity only increases as the number of address bits increases
- delay is only an issue as the number of bits and hence complexity increases

Mixed response to the question.

Average 2.09/4 (52%)

Q23. - PWN

This question tests your understanding of the two-dimensional addressing scheme, particularly its advantages over the one-dimensional scheme, which are:

- permits much larger memory structures due to the use of two decoders
- is simpler with regards to the size/complexity of the two decoders (easier to implement)
- possible speed improvements due to the reduction in complexity
- scalable due to the smaller sized decoders

Two sensible advantages required. I was not only looking for the disadvantage, but also an explanation of why.

This question is a bit of a giveaway as you will already have known one "disadvantage" of the one-dimensional scheme from the previous question.

General points:

- I didn't want a description of the operation or any disadvantages
- a lot of answers only provided one example, lacked detail or failed to discuss why.
- results in the use of fewer memory locations – no the number is the same for the same size address bus
- results in a smaller memory … not necessarily, as the memory cells themselves will be the same technology – the need to identify two lines going high may in fact increase the size of each memory location as you may need an AND gate to decide if the two lines are high
- can store more data with fewer bits/more data can be stored in the array … really?

Overall, the question was poorly answered.

Average 0.93/2 (47%)

Q24. - PWN

This question tested your understanding of the two-dimensional addressing scheme and how you select the decoder size.

The address bus is 8-bits, so can be implemented using two 4-to-16 line decoders. I was not asking for the array size for the two-dimensional scheme. Generally, any incorrect answers were due to calculating the wrong size decoders.

Average 0.60/1 (60%)

Q25. - PWN

So, this question required you to write some code! Just the contents of the always block for the decoder design given.

It requires you to understand the operation of the 3 to 8 line decoder. That is the corresponding bit of the 8-bit output goes high (the rest are zero) according to the value of the 3-bit binary input. So if the input is 000, then bit 0 of the output goes high, if 001, then bit 1, 010, then bit 2 etc.

The design could be implemented in a number of ways, the simplest being a case statement.

```
case(address_in)
  3’b000:  decode_out = 8’b0000_0001;
  3’b001:  decode_out = 8’b0000_0010;
  3’b010:  decode_out = 8’b0000_0100;
  3’b011:  decode_out = 8’b0000_1000;
  3’b100:  decode_out = 8’b0001_0000;
  3’b101:  decode_out = 8’b0010_0000;
  3’b110:  decode_out = 8’b0100_0000;
  3’b111:  decode_out = 8’b1000_0000;
  default: decode_out = 8’hX;
endcase
```

Marks are awarded as follows:

- correct use of case (syntax correct) including endcase – 1 mark
- inclusion of a default – 1 mark
- all 8 assignments correct – 2 marks (½ mark for each pair correct)
- no syntax errors – 1 mark

Generally, I found the question to be answered well. Marks were generally lost due to

- syntax errors (missing semicolons etc)
- missing endcase
- missing default
- incorrect assignments
- not using ‘b with binary numbers, otherwise 111 will be treated as “one hundred and eleven”
- Don’t copy the code given – there’s no need to!

Average 2.95/5 (59%)