Cover Page

Information

Department of Computer Science

Examination paper for TDT4258 Low Level Programming

Examination date: 14-December-2020

Examination time (from-to): 09:00 to 12:00

Permitted examination support material: A / All support material is allowed

Academic contact during examination: Rakesh Kumar
Phone: 91 26 21 63

Technical support during examination: Orakel support services
Phone: 73 59 16 00

OTHER INFORMATION

Make your own assumptions: If a question is unclear/vague, make your own assumptions and specify them in your answer. Only contact academic contact in case of errors or insufficiencies in the question set.
Saving: Answers written in Inspera Assessment are automatically saved every 15 seconds. If you are working in another program remember to save your answer regularly.
Cheating/Plagiarism: The exam is an individual, independent work. Examination aids are permitted, but make sure you follow any instructions regarding citations. During the exam it is not permitted to communicate with others about the exam questions, or distribute drafts for solutions. Such communication is regarded as cheating. All submitted answers will be subject to plagiarism control. *Read more about cheating and plagiarism here.*

Notifications: If there is a need to send a message to the candidates during the exam (e.g. if there is an error in the question set), this will be done by sending a notification in Inspera. A dialogue box will appear. You can re-read the notification by clicking the bell icon in the top right-hand corner of the screen. All candidates will also receive an SMS to ensure that nobody misses out on important information. Please keep your phone available during the exam.

Weighting: There are 10 questions in the exam and each question is worth 5 points.

ABOUT SUBMISSION

Your answer will be submitted automatically when the examination time expires and the test closes, if you have answered at least one question. This will happen even if you do not click “Submit and return to dashboard” on the last page of the question set. You can reopen and edit your answer as long as the test is open. If no questions are answered by the time the examination time expires, your answer will not be submitted.

Withdrawing from the exam: If you become ill, or wish to submit a blank test/withdraw from the exam for another reason, go to the menu in the top right-hand corner and click “Submit blank”. This cannot be undone, even if the test is still open.

Accessing your answer post-submission: You will find your answer in Archive when the examination time has expired.

1 Assembly programming

Please write the ARM assembly code corresponding to the following C code:

```
int simple_calc (int x, int y)
{
   return (x - y);
}

int calculate (int a, int b, int c, int d)
{
   if (a > b)
   {
      return (c + d);
   }
   else
   {
      return simple_calc(c, d);
   }
}

void main()
{
   int a = 10, b = 20, c = 100, d = 200;
   int d = calculate(a, b, c, d);
}
```

Your assembly code should contain code and data sections. Please comment your code appropriately. (Please translate the code as it is. That means do NOT apply any optimizations such as constant propagation, constant folding, dead code elimination etc.)

Fill in your answer here

Format

Words: 0

Maximum marks: 5

2 DMA

Please explain what Direct Memory Access (DMA) is. What benefit does it provide? Where can the processor get data for calculations while DMA is in operation?

Fill in your answer here

Format

Words: 0

Maximum marks: 5

3 Processor Design

Please explain the difference between a **single-cycle** and **pipelined** processor design. What are the factors that can prevent a 3-stage pipelined processor from achieving 3x (three times) performance gain over single-cycle processor design?

Fill in your answer here

Format

Words: 0

Maximum marks: 5

4 Caches

Consider a **4-way set-associative cache** that uses **20-bits for tag, 6-bits for index, and 6-bits for byte offset**. Find the cache size (capacity) in bytes. Please elaborate how you calculated the cache size.

Fill in your answer here

Format

Words: 0

Maximum marks: 5

5 Virtual Memory

Please elaborate how virtual memory solves **capacity** and **safety** issues in memory systems.

Fill in your answer here

Format

Words: 0

Maximum marks: 5

6 Memory Regions

A program typically needs four memory regions for its execution: **Stack, Heap, Static Data,** and **Instructions/Code**. What is stored in each of these regions? Why do Stack and Heap grow in opposite directions (towards each other) rather than in the same direction?

Fill in your answer here

Format

Words: 0

Maximum marks: 5

7 C programming

In the code below, implement the *update_bit* function with three input arguments *numA*, *numB*, and *pos*. The function checks if the bit at position *pos* in number *numA* is set. If the bit is set, it toggles the bit at position *pos* in *numB*; otherwise it does not do anything. Finally, the function returns the value of *numB*.

Position is zero-based and starts from the right. The position of the least significant bit is 0.

For example, the bit on the position 3 in 9 (1001 in binary) is 1. Therefore, *update_bit (9, 0, 3)* should return 8 and *update_bit (9, 0, 2)* will return 0.

Please comment you code appropriately.

```
int update_bit (int numA, int numB, int pos)
{
   //Your code goes here
}

void main()
{
   printf("%d", update_bit (9, 0, 3));
}
```

Fill in your answer here

```
1
```

Maximum marks: 5

8 Compilers

Apply these three optimization operations in order:

1. Dead code elimination
2. Constant propagation
3. Constant folding

to the following piece of C code **repeatedly until no more optimizations are possible**. Show the resulting code and indicate the modified and/or eliminated operations **after each optimization step**.

```
int q = 20;
int r = 11 - (q / 5);
int s;
s = r * 5;
if (s > 13) {
  s = r + 14;
}
return s * (100 / q);
```

Fill in your answer here

```
1
```

Maximum marks: 5

9 Operating Systems

The most simple operating system process model consists of the three process states **running**, **ready**, and **blocked** and related transitions between these states.

Describe **all** the transitions between these process states that take place when:

1. A process requests a long-running I/O operation from the operating system
2. A long running I/O operation completes and notifies the operating system of its completion
3. A timer interrupt occurs

For each transition, indicate in a few words **why** this transition occurs.
If there are multiple different options for transitions that can occur, please describe all of them.

Fill in your answer here

Format Σ

Words: 0

Maximum marks: 5

10 Power consumption

Please explain how **Clock Gating** and **Power Gating** work. What type of power consumption do they reduce?

Fill in your answer here

Format

Words: 0

Maximum marks: 5