

Exam 2021-06-03

⚠ Det här är en förhandsvisning av den publicerade versionen av quizet

Startad: 2 dec kl 8.39

Instruktioner för Quiz

Allowed help: You can use course material but you are not allowed to communicate with anyone except the TAs and the teacher. You can reach us at: stefanos.kaxiras@it.uu.se (<mailto:stefanos.kaxiras@it.uu.se>); it-avdark-ta@lists.uu.se (<mailto:it-avdark-ta@lists.uu.se>).

You can call in using zoom at: <https://uu-se.zoom.us/j/69653005890> ➞ (<https://uu-se.zoom.us/j/69653005890>)

In case of an emergency you can call at 070-4250394.

Language: Answers in English (Swedish can be used if needed, but try to have the main points of your answer also in English)

General information

- There are 19 questions in this exam.
- The maximum score is 72p.
- You need 36p to pass (this corresponds to earning all 12 bonus points – each **bonus** question gives you 3p on a specific question).
- You will automatically be given maximum points (3p) for the corresponding bonus questions if you have already received the corresponding bonus for the **Participation**, **Lab**, or **Hand-in**. You will not have to provide any answers for that question but clearly state that for such a question you claim the bonus you have already received.



In case you think that you have earned a bonus point that is not accounted for, we suggest that you solve that question (just to be safe), but state in the solution that you think you should have received a bonus point. We will sort that out later...

Grades

36p – 47p -> grade 3


48p – 60p -> grade 4

61p – 72p -> grade 5

Formatting your Answers

Most questions are essay questions. In these questions, please write your answers directly in Studium and indicate the sub-part ("a.", "b.", "c.", etc.) of the questions (if applicable) that you are

answering.

If you need to include a file (such as a drawing), please attach a document (Insert > Document > Upload document). Preferably use PDF, PNG, or Microsoft office formats, though other formats (found [here](https://community.canvaslms.com/t5/Instructor-Guide/What-types-of-files-can-be-previewed-in-Canvas/ta-p/607) ) are supported as well.

Not readable and/or not understandable answers result in zero points.

If any of the questions seem unclear, state your interpretation and assumptions clearly.

Note that some of the questions have several sub questions (i.e. “*What is ‘X’ **and** how does it improve ‘Y’...*”). Please answer all of them to get the full score.

Good luck!

// Stefanos, Per, Chris

Fråga 1

3 poäng

(If you have the **Lab 1 bonus point**, you may skip this question and still receive full mark.)

Blocking can be used to improve the effectiveness of a cache.

- Give a code example of how this technique is applied (first un-optimized and then optimized). **(2p)**
- What single locality property (name the important one) is primarily improved using this scheme? **(1p)**



p



0 words



Fråga 2

3 poäng

(If you have the **Hand-in 1 bonus point**, you may skip this question and still receive full mark.)

- a. Make a drawing (Note: You can upload files using the menu above the answer text box) of a **physically addressed cache** with the following sizes. Clearly show the address bits ranges used for the indexing, the comparisons, and multiplex ("mux") selects, etc. Describe, either in words or in logic, the functionality of all logic needed to resolve a read access to the cache. **(2p)**
- Cache size: 256 Kbyte
 - Cache line size: 32 byte
 - Organization: 4-way
 - CPU word size: 8 byte (this is the size of the data unit always delivered from the cache to the CPU)
 - Physical address size: 48 bits
- b. Which bit ranges would change if the cache size was increased to 1 Mbyte while all the other parameters stayed the same? State the new bit ranges (you do not need to make completely new drawing) **(1p)**



p



0 words



Fråga 3

3 poäng

(If you have the **Participation 1 bonus point**, you may skip this question and still receive full mark.)

- a. What is the purpose of the TLB in address translation? **(1p)**
- b. Describe how a lookup is performed in a TLB (about 4 sentences). **(1p)**
- c. What information is stored in a TLB entry? **(1p)**



p



0 words



Fråga 4

3 poäng

(If you have the **Lab 2 bonus point**, you may skip this question and still receive full mark.)

- What is the advantage of a *ticket-based* lock compared with a *Test and Test&Set* spinlock? **(1p)**
- Write the pseudo-code for barrier synchronization, assuming that you have access to the synchronization primitives `lock(L)` and `unlock(L)`. Include comments describing the purpose of the operations. **(2p)**

p



0 words



Fråga 5

3 poäng

(If you have the **Hand-in 2 bonus point**, you may skip this question and still receive full mark.)

Dekker's algorithm is described by the following parallel pseudo-code:

Initially:

```
A := 0;  
B := 0;
```

Thread 0:

```
A := 1;  
if (B==0) then  
  print("Thread 0 won");
```

Thread 1:

```
B := 1;  
if (B==0) then  
  print("Thread 0 won");
```

- What feature is implemented by Dekker's algorithm? (I.e., what did Dekker use it for?) **(1p)**
- Under which of the three memory models: *sequential consistency*, *total store order* and/or *weaker consistency*, would it work as Dekker anticipated? **(1p)**
- How would the code be changed in order to work for weaker models? **(1p)**

p



0 words

**Fråga 6****3 poäng**

(If you have the **Participation 2 bonus point**, you may skip this question and still receive full mark.)

Draw the two state transition diagrams (one for snooping the bus (*) and one for the CPU accessing the cache (**)) for an idealized MOSI cache coherence protocol where each state transition and output signal happens “*atomically*”. For each state transition arc between the coherence states, mark the input signal (what caused this transition)/output action (e.g., is Data provided by this cache?)

(*) Snooping transactions: ReadToShare (RTS), ReadToWrite (RTW), Invalidate (INV), Writeback (WB).

(**) CPU accesses: CPUread, CPUwrite, CPUreplacement

p



0 words



Fråga 7

3 poäng

(If you have the **Lab 3 bonus point**, you may skip this question and still receive full mark.)

- What is false sharing and under which of the parallel programming paradigms OpenMP, MPI and/or Pthreads, could an application typically experience such a problem? **(2p)**
- How can you remove false sharing from an application? **(1p)**

p



0 words

**Fråga 8****3 poäng**

(If you have the **Hand-in 3 bonus point**, you may skip this question and still receive full mark.)



- Draw the structure (Note: You can upload files using the menu above the answer text box) of a Dir_3B directory assuming that it tracks the coherence of 56 caches. Clearly show how many bits each field has. **(2p)**
- Describe the difference between coherence protocols based on a Dir_3B directory and a Dir_3NB directory. **(1p)**

p



0 words



Fråga 9

3 poäng

(If you have the **Participation 3 bonus point**, you may skip this question and still receive full mark.)

A 2-way SMT pipeline and a 2-way super-scalar pipeline can both start the execution of two instructions each cycle. State how they differ in terms of:

- How they select the instructions to execute. **(1p)**
- What hardware they need to replicate compared with a 1-way implementation. **(1p)**
- What kind of *parallelism* they exploit. **(1p)**

Redigera Visa Infoga Format Verktyg Tabell

12pt ▾

Paragraph ▾

B*I*UA ▾T² ▾

p



0 words



Fråga 10

3 poäng

(If you have the **Lab 4 bonus point**, you may skip this question and still receive full mark.)

- According to Flynn's taxonomy, what kind of parallelism is primarily explored with the SSE, MMX and AVX instructions of x86? (write both the four-letter acronym and its word-by-word meaning) **(2p)**
- What is the other common parallelism expressed in Flynn's taxonomy (write both the four-letter acronym and its word-by-word meaning) and how does it differ from 10a? **(1p)**

Redigera Visa Infoga Format Verktyg Tabell

12pt ▾

Paragraph ▾

B*I*UA ▾T² ▾

p



0 words



Fråga 11

3 poäng

(If you have the **Hand-in 4 bonus point**, you may skip this question and still receive full mark.)

There are three different classes of hazards: *Structural hazards*, *data hazards* and *control hazards*.






Please state which of them (a single one) is mainly removed by:

- a. Branch prediction? **(1p)**
- b. Register renaming? **(1p)**
- c. Superscalar architecture (compared with single-issue architectures)? **(1p)**

Redigera Visa Infoga Format Verktvg Tabell

12pt ∨ Paragraph ∨ | **B** *I* U A ∨  ∨ T² ∨ | :

p

  | 0 words |   



Fråga 12

3 poäng

(If you have the **Participation 4 bonus point**, you may skip this question and still receive full mark.)

- a. Explain the most fundamental technique used by GPUs to hide memory latency? How does it differ from the techniques used by CPUs? **(2p)**
- b. Name two popular "high-level languages" that are commonly used to program GPUs today. **(1p)**

Redigera Visa Infoga Format Verktyg Tabell

12pt ▾ Paragraph ▾ | **B** *I* U A ▾  ▾ T² ▾ | :

p



0 words



All the three RISC CPUs (e.g. cores in a multicore) in a **MOSI** shared-memory sequentially consistent multiprocessor execute the following code:

CPU1	CPU2	CPU3
<pre>A = 1; while (C == 0) { };</pre>	<pre>while (A == 0) { };</pre> <pre>A = 2;</pre> <pre>B = 1;</pre>	<pre>while (B == 0) { };</pre> <pre>C = A;</pre>



The CPUs execute ALU instructions (register-to-register) and memory instructions, moving values between memory and registers. There are two kinds of memory instructions in this example: **LD** and **ST**.

CPU3 will start slightly ahead of the others and will execute its first memory instruction before **CPU2** starts its first memory instruction. **CPU1** will start last. Then, the CPUs will take turns and execute one memory instruction each (this is an artificial constraint for exam purposes only).

Initially **A**, **B** and **C** reside only in memory and have the values **0**. They belong to different cachelines.

After the parallel execution is done for all of the CPUs, the cachelines that are still

in the caches will get replaced by data brought in by some other execution. This activity should also be reflected in your answer.

The following four bus transaction types can be seen on the snooping bus connecting the CPUs (cores):

- **RTS**: ReadtoShare (reading the data with the intention to read it)
- **RTW**: ReadToWrite (reading the data and requesting write permission)
- **WB**: Writing data back to memory
- **INV**: Invalidating other caches copies

The table that follows (in the next question) shows every state change and/or value change of A, B and C in each CPU's cache **according to one correct interleaving of the memory accesses**. For each line, the bus transaction that occurs on the bus (if any) as well as which device is providing the corresponding data (if any) are shown.

Unfortunately for you the coherence protocol in this solution is fairly “broken” and introduces a few bugs compared to a proper-functioning protocol. Some (but not all) bus transactions, states, values, and data-providers are wrong. You must find all the bugs and correct them. Mark in red (or strike out) the incorrect entry and write beneath it the correct one (make it clear). For convenience, you should be able to copy and paste the table into your answer and then modify it. Alternatively, you can use the table tool in the text box to create a new table, as long as you keep the columns the same. If you run into problems with the text editing on Studium it is also possible to upload a file (e.g., PDF or Word) with your answer. **Please fill in your answer in the question that follows.**

Grading: There are **20 bugs** in the table. If you find them all you get 6 points. But, you will lose 0.5 point for each correct entry you mistakenly mark as a bug. You will not get negative total points, regardless of how many mistakes you make.



Fråga 13

6 poäng

Here is the table from the previous question. Please answer the previous question here. Studium allows you to copy the table from the answer into the question form, but feel free to recreate the table if it is more convenient for you. If you run into any issues with tables on Studium, it is also possible to attach a file (e.g., PDF or Word) with your answer.

CPU action	Bus Transaction (if any)	State/value after the CPU action									Data is provided by
		CPU1			CPU2			CPU3			[CPU 1, 2, 3 or Mem] (if any)
		A	B	C	A	B	C	A	B	C	
Initially		I	I	I	I	I	I	I	I	I	
CPU3: LD B	RTS B								S/0		Mem
CPU2: LD A	RTS A				S/0						Mem
CPU1: ST A (1)	RTW A	M/1			I						Mem
CPU3: LD B	RTS B								S/0		-
CPU2: LD A	RTS A	O/1			S/1						CPU1
CPU1: LD C	RTS C			S/0							Mem
CPU3: LD B	RTS B								S/0		-
CPU2: ST A (2)	RTW A	I			M/2						CPU1
CPU1: LD C	RTS C			S/0							-
CPU3: LD B	RTS B								S/0		-
CPU2: ST B (1)	RTW B					M/1			I		Mem
CPU1: LD C	RTS C			S/0							-
CPU3: LD	RTS B					O/1			S/1		CPU2

B											
CPU1: LD C	RTS C			S/0							-
CPU3: LD A	RTS A				M/2			S/2			CPU2
CPU1: LD C	RTS C			S/0							-
CPU3: ST C (A)	RTW C			I						M/2	Mem
CPU1: LD C	RTS C			S/2						O/2	CPU3
...											
CPU1: Repl C	WB C			S/2							CPU1
CPU2: Repl A	WB A				S/2						CPU2
CPU2: Repl B	WB B					S/1					CPU2
CPU3: Repl A	-							S/2			-
CPU3: Repl B	-								S/1		-
CPU3: Repl C	-									S/2	-

RedigeraVisaInfogaFormatVerktygTabell



12pt ▾Paragraph ▾|**B***I*UA ▾ ▾T² ▾|⋮

p



0 words



Fråga 14

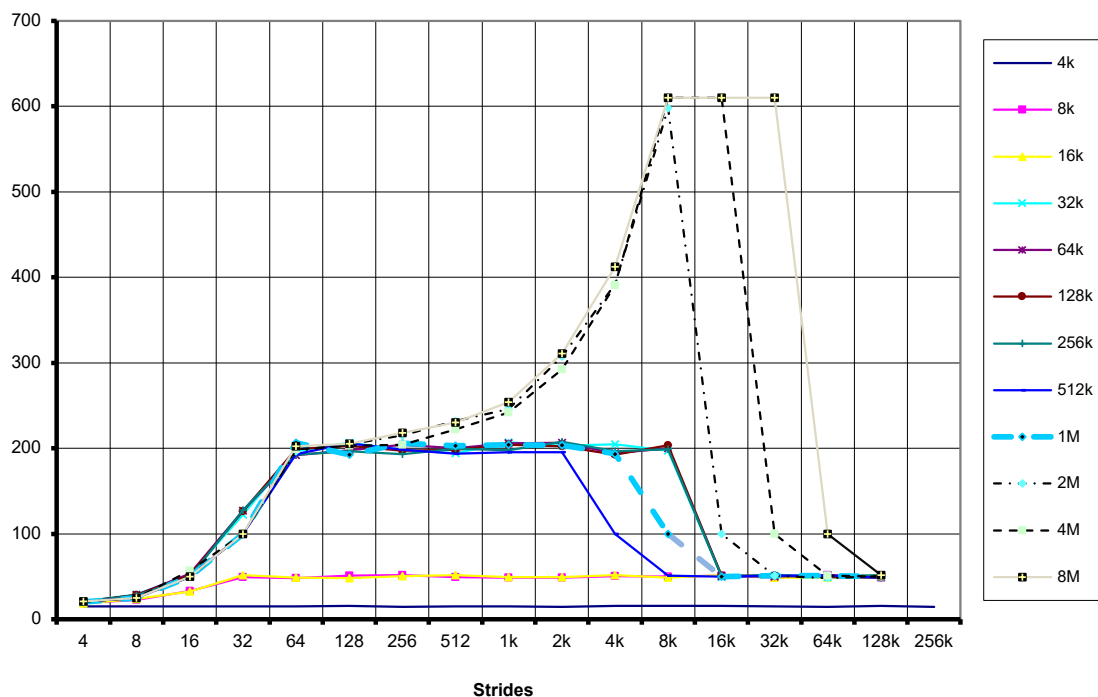
6 poäng

A simple microbenchmark code below is used to draw curves that characterizes the properties of the underlying memory system. We assume that both the caches and the TLB are fully associative.

```
for (times = 0; times < Max; time++)          /* many times*/
  for (I = 0; i < ArraySize; i = i + Stride)
    dummy = A[ i ];                          /* touch an item in the array */
  /*
```

The curve below shows the average access time to a memory system as a function of Stride for different ArraySizes.

Microbenchmark



- What is the cache line size for the 1st level cache?
- What is the size of the 1st level cache?
- What is the cache line size and cache size for the 2nd level cache?
- What is the size of the 2nd level cache?
- What is the page size?
- How many entries does the TLB have?

Notes:

- The 4kB line is at the bottom of the plot.
- Above it are the 8kB - 16kB lines.
- **Write all the answers using the appropriate prefix and unit, e.g., "16kB" or "1MB" (for cache sizes), "16" (for cache line sizes), "12" (for TLB entries), etc. Always use the highest prefix possible, e.g., write "8k" instead of "8192". 1024B = 1kB and 1024kB = 1MB.**

Fråga 15**4 poäng**

(This is the **first** question in a series of three.)

The following three-part problem combines branch-prediction (part 1), register renaming (part 2), and out-of-order scheduling (part 3).

Consider the code below, it *increments non-zero* values of , a 4x4 matrix of doubles (8-bytes). (An image of the matrix, before and after running the code, can be seen in the second part of the question series.)

```
for (i = 0; i < 4; i++) // rows
  for (j = 0; j < 4; j++) { // columns
    if (x[i,j] != 0)
      x[i,j] += 1
  }
```

In this simple pseudo assembly code that corresponds to the program, show the RAW instruction dependencies (dependencies for the first two instructions are already filled). Because of the multiple branches an operand may be produced by

different instructions depending on the path taken. In this case, show all dependencies.

# (label:)	Instruction	Comment	RAW Dependencies
1: loop:	MULI R3, R1, #4	R3 = i*4	10: (R1)
2:	ADD R4, R2, R3	R4 = j+i*4	1: (R3), 7: (R2), 9: (R2)
3:	LDD R5, x(R4)	R5 = x[i,j]	
4:	BNEQZ R5, cont	R5 != 0 ? cont	
5:	ADDI R5, R5, #1	R5 += 1	
6:	STD R5, x(R4)	x[j+i*4] = R5	
7: cont:	ADDI R2, R2, #8	j += 1	
8:	BLTEQ R2, #4, loop	j <= 4 ? loop	
9:	ADDI R2, Rzero, #0	j = 0	
10:	ADDI R1, R1, #8	i += 1	
11:	BLTEQ R1, #4, loop	i <= 4 ? loop	

For instructions 3-11, indicate what other instructions (if any) they have a RAW dependence on and through what register.

(You may for example copy the pseudo assembly code above and fill in the dependence column or create your own table.)

Redigera Visa Infoga Format Verkttyg Tabell

12pt ▾ Paragraph ▾ | **B** *I* U A ▾  ▾ T² ▾ | :

p



0 words



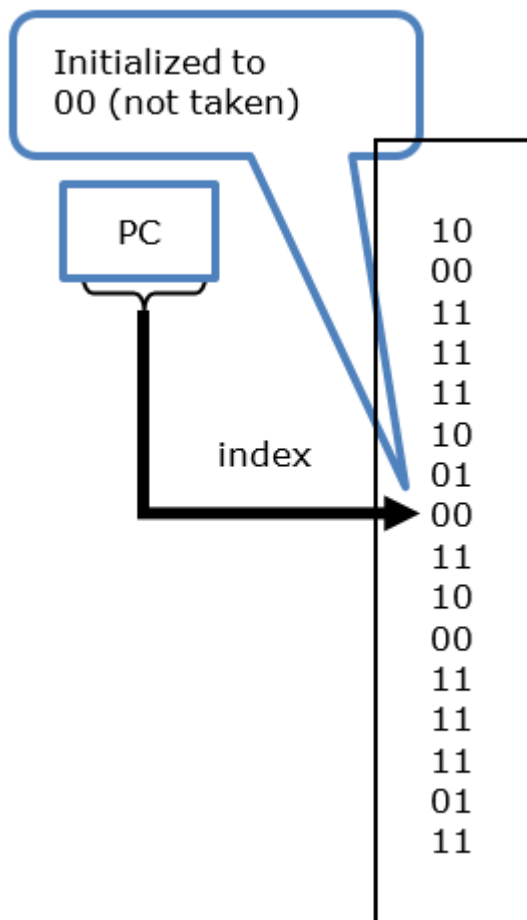
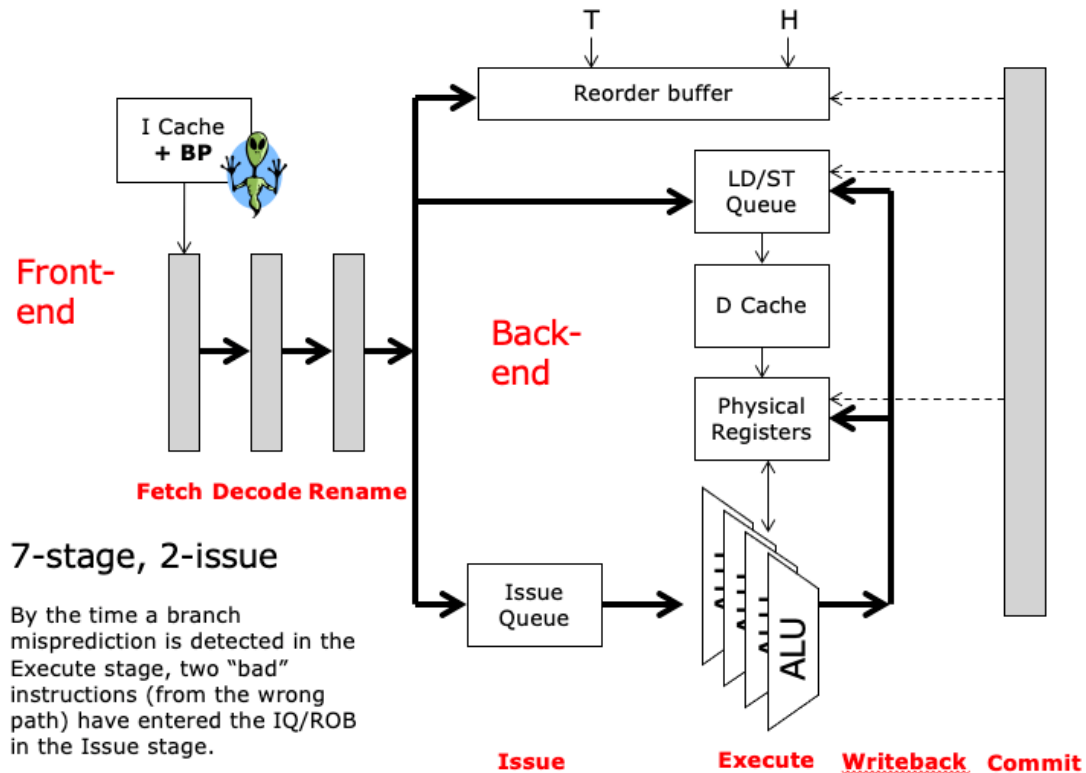
Fråga 16

6 poäng

*(This is the **second** question in a series of three.)*

Assume that the code (from the previous part of the question series) is going to run in the 7-stage, 2-way, superscalar, dynamically scheduled (out-of-order) core shown below. A **2-bit branch predictor** (shown below) is used to predict branches. For any branch instruction assume that the initial value of the corresponding predictor entry is 00. Branches are predicted as Not-Taken (NT) if the value of the predictor is 00 or 01 and as Taken (T) if the value of the predictor is 10 or 11.



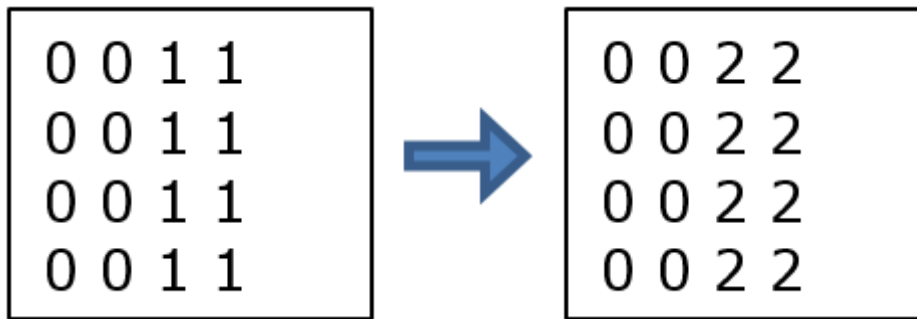


For each miss-predicted branch, the first two instructions from the wrong path of execution make it into Issue Queue (IQ) and the Reorder Buffer (ROB).

Show the sequence of instructions that enter the IQ/ROB for the first five iterations of the inner loop body:

- Using the 2-bit branch predictor discussed earlier, show the prediction for each branch instruction next to it: “NT” for Not-Taken and “T” for Taken. On the rightmost three columns, keep track of the branch predictor state for the three branches in the code (4:, 8:, 11:)
- Assume that if the branch is miss-predicted, 2 instructions from the wrong path enter the IQ/ROB after the branch and before the correct instruction from the correct path.
- If the prediction is incorrect (i.e., a miss-prediction) then cross it out along with the two instructions from the wrong path that follow. It is OK to leave empty cells in a row after an iteration ends, or after Taken branches so you can align the instructions across iterations.

Consider the case where the code uses the matrix on the left as input (yielding the matrix on the right as output):



Below is a table that you can copy and complete filling (the first iteration and the beginning of all the rest is already filled), alternatively you can create your own table.

Iteration	Instruction fetch sequence (use instruction labels from Part 1)														New BP state		
	Predicted instruction: 4: 8: 11:														4:	8:	11:
	Initial state: 00 00 00														00	00	00
1)	1:	2:	3:	4:NT	5:	6:	7:	8:NT	9:	10:					01	01	00
2)	1:	2:	3:														
3)	1:	2:	3:														
4)	1:	2:	3:														
5)	1:	2:	3:														

Redigera Visa Infoga Format Verktyg Tabell

12pt ▾ Paragraph ▾ | **B** *I* U A ▾  ▾ T² ▾ | ⋮

p



0 words

**Fråga 17****6 poäng***(This is the **third** question in a series of three.)*

Assume now that we have **perfect branch prediction** and the sequence of instructions that enters the IQ/ROB is always from the correct path of execution (i.e., the branches are always followed by the correct next instruction and no instruction is ever squashed). Do the following:

1. In the first table below, show the instructions of the first two iterations in the IQ (copy the instructions from Part 2 -- the previous question -- but leave out the extra squashed instructions that follow any mispredicted branches)
2. In the same table, use register renaming to assign physical registers to the architectural registers and replace architectural registers everywhere else it is needed, as shown for the first four instructions of the IQ that are already register-renamed.
3. In the second table, schedule instructions in the IQ for a 2-way out-of-order superscalar **assuming that all LDD instructions have a latency of 4 cycles** (if the LDD is in cycle n , its result can be used by an instruction in cycle $n+5$), as shown for the four first instructions in the IQ that are already scheduled.

Show the IQ and renaming here:

Instruction sequence number in the IQ	Instructions	Register Renaming:
		(from before: R1→Px R2→Py)
IQ1	1: MULI R3→P1, Px, #4	R3→P1
IQ2	2: ADD R4→P2, Py, P1	R4→P2
IQ3	3: LDD R5→P3, x(P2)	R5→P3
IQ4	4: BNEQZ P3, cont	--
IQ5		
IQ6		
IQ7		
IQ8		
IQ9		
IQ10		
IQ11		
IQ12		
IQ13		
IQ14		



IQ15		
IQ16		

Schedule your instructions here:

Cycle:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Issue1	IQ1	IQ2	IQ3					IQ4							
Issue2															

It should be possible on Studium to copy the tables from this question into your answer and edit it there. If you run into any issues you can also re-create the table in a separate document (e.g., in Word) and upload a document for your answer (e.g., Word or PDF).

Redigera Visa Infoga Format Verktyg Tabell

12pt ▾ Paragraph ▾ | **B** *I* U A ▾  ▾ T² ▾ | ⋮

p

  | 0 words |   ⋮

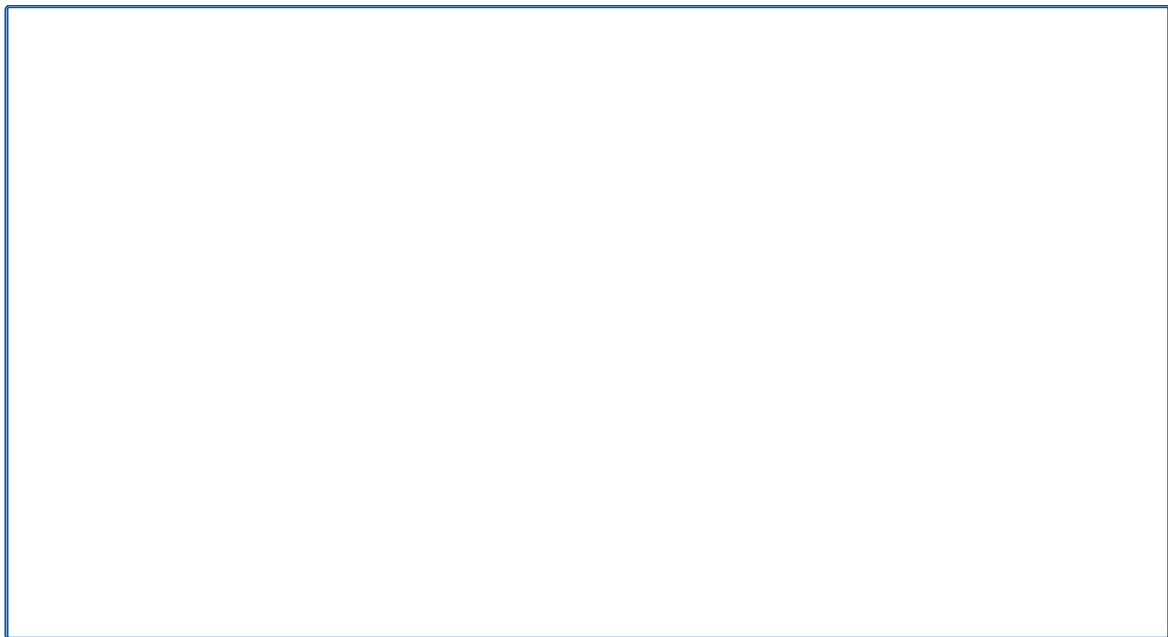
The following code contains a simple lock implementation using a Test-and-Set atomic operation:

```
void lock(lock_variable) {  
    while true {  
        if (TAS(lock_variable) == 0) break;  
    }  
}  
  
void unlock(lock_variable) {  
    lock_variable = 0;  
}
```

- a. What is the problem with such a simple lock? **(2p)**
b. Describe (including pseudocode) how the lock can be made better. **(2p)**

Redigera Visa Infoga Format Verktyg Tabell

12pt ▾ Paragraph ▾ | **B** *I* U A ▾  ▾ T² ▾ | :



p



0 words



Fråga 19

4 poäng

GPU vendors often compare the performance of their GPUs vs. high performance CPUs. You might see claims such as “Our GPU is 30x faster than the latest Intel CPU”. As a user of such GPUs, do you expect to reach such performance in practice? Why or why not? Write 3-4 sentences.

Redigera Visa Infoga Format Verktvg Tabell

12pt ▾ Paragraph ▾ | **B** *I* U A ▾  ▾ T^2 ▾ | :

p



0 words



Quiz sparad kl. 8.40

Lämna in quiz