Set word on a lab by processor.

Compare how the given instruction

(b) How does the result of a)

Set for an advanced processor.

Instructions to make this instruction

(a) Use register renaming and predicating

```
ADD $x1, $x5, $16
STUR $x1, ($x6, $8)
```

```
ADD $x1, $x5, $x0
BLT NEXT
```

```
CMP $x6, $x1
SUBI $x6, $x5, $8
LDUR $x5, [x0, $16]
```
Write assembly code that correctly add two numbers.

There are two 128-bit numbers stored into four registers.

Num 2: X2 X3
      0   127

Num 1: X0 X1
      0   127
Below is a table of a cache system. Answer the following questions based on this information.

<table>
<thead>
<tr>
<th>Hit Rate</th>
<th>100,000 cycles</th>
<th>Disk</th>
<th>Main Memory</th>
<th>L2</th>
<th>L1</th>
<th>Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>100%</td>
<td>80 cycles</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>99%</td>
<td>20 cycles</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>70%</td>
<td>1 cycle</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>60%</td>
<td>Hit Time</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(Note: Disk is not a valid level since it has a 100% hit rate.)
LDR R9, [X'41', #0]
LDUR X9, [X'41', #0]

ADDI X9, X5, #1
LSL X9, X9, #2

STUR X9, [X'41', #0]

Convert this assembly program to machine code.
machine.

Draw the datapath for the
instructions from lecture.
Instruction as well as all other
“CSEL”
Design a single-cycle CPU that
<table>
<thead>
<tr>
<th>Level</th>
<th>Hit Time</th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>Hit Time</td>
<td>25%</td>
<td>50%</td>
<td>75%</td>
</tr>
<tr>
<td>Mid</td>
<td>Hit Time</td>
<td>1st</td>
<td>2nd</td>
<td>3rd</td>
</tr>
<tr>
<td>High</td>
<td>Hit Time</td>
<td>1st</td>
<td>2nd</td>
<td>3rd</td>
</tr>
</tbody>
</table>

Which one would cause the least loss of efficiency?

If you have to show down any levels hit-time by a factor of 2.
A.

B.

Given a 2-way set associative cache with 2 lines of 1 byte blocks, show an access pattern that will result in a capacity miss (A) and an access pattern that will result in a conflict miss (B). The given cache is shown below.

Explain the difference between a capacity miss and a conflict miss.
<table>
<thead>
<tr>
<th>Level</th>
<th>Hit Rate</th>
<th>Hit Time</th>
<th>Main Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>L2</td>
<td>85%</td>
<td>2 cycle</td>
<td>100</td>
</tr>
<tr>
<td>L1</td>
<td></td>
<td></td>
<td>100</td>
</tr>
<tr>
<td>Disk</td>
<td></td>
<td></td>
<td>5000</td>
</tr>
</tbody>
</table>

Which is the best L2 cache for this system?
How accurate is a 8-bit predictor?

How accurate is a 4-bit predictor?

T, T, T, N, T, T, N, N

Given branch pattern:
| No-Op | Start x4, cx5, #o | No-Op |
| No-Op | Load x4, cxs#o |
| No-Op | Store x0, cx5, #o |
| No-Op | No-Op |
| No-Op | No-Op |
| No-Op | No-Op |
| No-Op | No-Op |
| No-Op | No-Op |

**Cycle 1:**

- ALU: Load Start
- Load/Store: Load

**Cycle 2:**

- ALU: No-Op

**Cycle 3:**

- ALU: No-Op

- SBRI x1, x1, #16 |
- ADD x0, x1, #16 |
- ADD x4, x6, x3 |
- SUB x0, x1, #16 |

**Cycle 4:**

- ALU: No-Op

**Cycle 5:**

- ALU: No-Op

**Cycle 6:**

- ALU: No-Op

**Cycle 7:**

To draw a conceptual diagram of the instruction set and the associated registers and memory, the following code is structured on a single line:

```
for all & instruction, let the ALU & load/store, and the other slots on a dual-way LWIM with 1 slot for no matter what. Schedule it and do not write-out one execution instruction(s) per line to a branch with branch delay slots (e).
```
B.LT LOOP
CMP X4, X0
SUBI X1, X1, #4
ADDS X4, X4, X3
LDUR X4, [X5, #0]
STRR X0, [X3, #0]
ADDI X0, X0, X2
LDUR X0, [X1, #0]
LOOP:

HW8a Question:

Draw the constraint graph of the following assembly code, assuming there is no delay slot. Then optimize the code to allow it run in a delay slot CPU.