This slide set is intended to give you some practice with problems that are similar to those that will appear on the test. Try to work out all of the problems before looking at the solutions, which are on the slide right after the problem statement.
Create a datapath for the `sub` instruction.

```
sub $rd, $rs, $rt
```
Create a datapath for the sub instruction.

Now, add in the bit width values.
Create a datapath for the `sub` instruction.

Now, add in the bit width values.

Which of the following control lines need to be set: RegWrite, MemWrite, MemRead?
Create a datapath for the `sub` instruction.

Now, add in the bit width values.

Which of the following control lines need to be set?
- `RegWrite` – Set!
- `MemWrite` and `MemRead` are not set.
Create a datapath for the $lw$ instruction.

$lw \ $rt, \ \text{immed}($rs$)$
Create a datapath for the \texttt{lw} instruction.

Now add in the bit width values.
Create a datapath for the \texttt{lw} instruction.

Now add in the bit width values.

Which of the following control lines need to be set: RegWrite, MemWrite, MemRead?
Create a datapath for the \texttt{lw} instruction.

Now add in the bit width values.

Which of the following control lines need to be set?
- RegWrite – Set!
- MemWrite – Unset!
- MemRead – Set!
Consider the following monstrous logic equation:

\[ D = (\overline{A} \cdot \overline{B} \cdot \overline{C}) \cdot (A \cdot B \cdot C) \cdot \left( (A \cdot B \cdot C) + (\overline{A} \cdot B \cdot C) + (A \cdot \overline{B} \cdot C) \right) \]

Fill in the truth table for this equation.
Consider the following monstrous logic equation:

\[ D = (\overline{A} \cdot \overline{B} \cdot \overline{C}) \cdot (A \cdot B \cdot C) \cdot ((A \cdot B \cdot \overline{C}) + (\overline{A} \cdot B \cdot C) + (A \cdot \overline{B} \cdot C)) \]

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
Consider the following monstrous logic equation:

\[
D = (\overline{A} \cdot \overline{B} \cdot \overline{C}) \cdot (A \cdot B \cdot C) \cdot \left( (A \cdot B \cdot \overline{C}) + (\overline{A} \cdot B \cdot C) + (A \cdot \overline{B} \cdot C) \right)
\]

Express this equation in the canonical sum-of-products form.

\[
D = (\overline{A} \cdot \overline{B} \cdot C) + (\overline{A} \cdot B \cdot \overline{C}) + (A \cdot \overline{B} \cdot \overline{C})
\]

Now, try to draw the hardware for this logic equation.
Consider the following logic equation:

\[ D = (\overline{A} \cdot \overline{B} \cdot C) + (\overline{A} \cdot B \cdot \overline{C}) + (A \cdot \overline{B} \cdot \overline{C}) \]
Translate the following C code fragment into MIPS. Let $a$ and $b$ be associated with $t0$ and $t1$, respectively. Let the base addresses of A and B be associated with $s0$ and $s1$, respectively.

```c
if(a < b){
    for(i = 0; i < a; i++){
        A[i] = B[i]
    }
}
```
Translate the following C code fragment into MIPS. Let a and b be associated with $t0 and $t1, respectively. Let the base addresses of A and B be associated with $s0 and $s1, respectively. Let i be associated with $t7.

```c
if(a < b){
    for(i = 0; i < a; i++){
        A[i] = B[i]
    }
}
```

```mips
slt  $t2, $t0, $t1
beq  $t2, $zero, L1
addi $t7, $zero, 0
LP:
    slt  $t2, $t7, $t0
    beq  $t2, $zero, L1
    sll  $t3, $t7, 2
    add  $t4, $s1, $t3
    lw   $t4, 0($t4)
    add  $t5, $s0, $t3
    sw   $t4, 0($t5)
    addi $t7, $t7, 1
    j LP
L1:
```
The Classic CPU Performance Equation is:

\[ CPU \text{ Time} = \#\text{Instructions} \times CPI \times \text{Clock Period} \]

Alternatively,

\[ CPU \text{ Time} = \frac{\#\text{Instructions} \times CPI}{\text{Clock Rate}} \]

A program on a machine has a clock rate of 3 GHz, each instruction takes 2 clock cycles on average, and the execution time is 4 seconds. How many instructions were executed in this program?
The Classic CPU Performance Equation is:

\[ CPU \text{ Time} = \#\text{Instructions} \times CPI \times \text{Clock Period} \]

Alternatively,

\[ CPU \text{ Time} = \frac{\#\text{Instructions} \times CPI}{\text{Clock Rate}} \]

A program on a machine has a clock rate of 3 GHz, each instruction takes 2 clock cycles on average, and the execution time is 4 seconds. How many instructions were executed in this program?

Answer: \(6 \times 10^9\) instructions
FINAL THOUGHTS

Knowing how to approach each of these problems (or related problems) will get you pretty far on the exam, but you still need to be comfortable with everything on the topics list. Especially:

• The details of how the MIPS instruction set works. Do you know what each of the fields is for and how the field is constructed?

• Great Architecture Ideas and ISA Design Principles.

• Trends in implementation technology, the power wall, and the effect of these on today’s computers.

• How is high-level source code converted into a running process?

• Etc…