1. The designers of that handheld game system you did a little work on last week are still trying to decide on a processor for their system. The two processors up for consideration and are described in Table 1.

<table>
<thead>
<tr>
<th></th>
<th>Arithmetic</th>
<th>Control</th>
<th>Memory</th>
<th>Floating Point</th>
<th>Power/MHz</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processor A</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>4mW/MHz</td>
</tr>
<tr>
<td>Processor B</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>8mW/MHz</td>
</tr>
</tbody>
</table>

Three example games are being considered. These games use the processor differently and require different amounts of processing per frame.

<table>
<thead>
<tr>
<th></th>
<th>Game 1</th>
<th>Game 2</th>
<th>Game 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arithmetic</td>
<td>30%</td>
<td>15%</td>
<td>10%</td>
</tr>
<tr>
<td>Control</td>
<td>40%</td>
<td>15%</td>
<td>20%</td>
</tr>
<tr>
<td>Memory</td>
<td>15%</td>
<td>60%</td>
<td>30%</td>
</tr>
<tr>
<td>Floating Point</td>
<td>15%</td>
<td>10%</td>
<td>40%</td>
</tr>
<tr>
<td>IPF</td>
<td>1,000,000</td>
<td>300,000</td>
<td>600,000</td>
</tr>
</tbody>
</table>

a. (4 points) How many megacycles does each game need per frame on each processor?

Calculating CPI:

CPI_{1A} = 0.3 \cdot 4 + 0.4 \cdot 2 + 0.15 \cdot 2 + 0.15 \cdot 4 = 2.9
CPI_{2A} = 0.15 \cdot 4 + 0.15 \cdot 2 + 0.6 \cdot 2 + 0.1 \cdot 4 = 2.5
CPI_{3A} = 0.1 \cdot 4 + 0.2 \cdot 2 + 0.3 \cdot 2 + 0.4 \cdot 4 = 3.0
CPI_{1B} = 0.3 \cdot 1 + 0.4 \cdot 1 + 0.15 \cdot 3 + 0.15 \cdot 5 = 1.9
CPI_{2B} = 0.15 \cdot 1 + 0.15 \cdot 1 + 0.6 \cdot 3 + 0.1 \cdot 5 = 2.6
CPI_{3B} = 0.1 \cdot 1 + 0.2 \cdot 1 + 0.3 \cdot 3 + 0.4 \cdot 5 = 3.2

Cycles/Frame = Cycles/Instruction \cdot Instructions/frame
1 MCycle = 10^6 Cycles

<table>
<thead>
<tr>
<th>MCycles/Frame</th>
<th>Game 1</th>
<th>Game 2</th>
<th>Game 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processor A</td>
<td>2.9 \cdot 1.0 = 2.9</td>
<td>2.5 \cdot 0.3 = 0.75</td>
<td>3.0 \cdot 0.6 = 1.8</td>
</tr>
<tr>
<td>Processor B</td>
<td>1.9 \cdot 1.0 = 1.9</td>
<td>2.6 \cdot 0.3 = 0.78</td>
<td>3.2 \cdot 0.6 = 1.92</td>
</tr>
</tbody>
</table>
b. (4 points) What clock rate would each processor have to run at to provide at least 30 frames per second in all games?

Cycles/Sec = Cycles/Frame \cdot Frames/Sec

1,A: 2.9 MCPF \cdot 30 \text{ FPS} = 87 \text{ MHz}
2,A: 0.75 MCPF \cdot 30 \text{ FPS} = 22.5 \text{ MHz}
3,A: 1.8 MCPF \cdot 30 \text{ FPS} = 54 \text{ MHz}
1,B: 1.9 MCPF \cdot 30 \text{ FPS} = 57 \text{ MHz}
2,B: 0.78 MCPF \cdot 30 \text{ FPS} = 23 \text{ MHz}
3,B: 1.92 MCPF \cdot 30 \text{ FPS} = 57.6 \text{ MHz}

Processor A: 87.0 MHz  
Processor B: 57.6 MHz

c. (3 points) In general, processors consume power approximately proportional to their clock rate. Assuming the processors consume power at the ratio shown in the last column of Table 1, what is the maximum power each of the processors would need to meet the 30fps target across all games?

\[ \text{Power} = \text{Power/MHz} \cdot \text{MHz} \]

(recall, \( \text{Power} = \text{Energy} / \text{Time} \))

Processor A: 4 mW/MHz \cdot 87MHz = 348mW = 0.35W
Processor B: 8 mW/MHz \cdot 57MHz = 461mW = 0.46W

Processor A: 348 mW  
Processor B: 461 mW

2. (3 points) What are at least three functions of the Atari 2600’s TIA chip?

Yes:
Image synthesis, sound synthesis, collision detection, paddle knob inputs (paddle input acceptable), joystick trigger inputs (joystick input NOT acceptable), console switches/levers/buttons.

Incorrectly yes:
Timers: actually in 6532, but slides said TIA did this.

No:
Joystick Direction, Paddle buttons, memory : handled by 6532 RAM/IO chip
“adding color to the display”: What does this even mean?
microprocessor: that’s the 6507, not the TIA
palette switching: the TIA holds the palette, but the CPU has to handle switching them blanking, overscan: aspects of the NTSC signal, not specific functions of the TIA
3. (6 points) Write code that would perform the following calculation on each of the following architecture types. Use mnemonics of the forms OPERATION `dst srcA [srcB]` and OPERATION `addr [reg]`. If you need to use temporary memory locations, name them temp1, temp2, etc.

\[ A = \frac{X \times Y}{W + Z} \]

**memory-memory:**

- `MUL temp1 X Y`
- `ADD A temp1 W Z`
- `DIV A temp1 A`

**accumulator:**

- `LOAD W`
- `ADD Z`
- `STORE temp1`
- `LOAD X`
- `MUL Y`
- `DIV temp1`
- `STORE A`

**stack:**

- `PUSH X`
- `PUSH Y`
- `MUL`
- `PUSH W`
- `PUSH Z`
- `ADD`
- `DIV`
- `POP A`

**register-register:**

- `LOAD X r1`
- `LOAD Y r2`
- `MUL r3 r1 r2`
- `LOAD W r4`
- `LOAD Z r5`
- `ADD r6 r4 r5`
- `DIV r7 r3 r6`
- `STORE A r7`