6 SIMD [90 points]

We have two SIMD engines: 1) a traditional vector processor and 2) a traditional array processor. Both processors can support a vector length up to 16.

All instructions can be fully pipelined, the processor can issue one vector instruction per cycle, and the pipeline does not forward data (no chaining). For the sake of simplicity, we ignore the latency of the pipeline stages other than the execution stages (e.g, decode stage latency: 0 cycles, write back latency: 0 cycles, etc).

We implement the following instructions in both designs, with their corresponding execution latencies:

Operation	Description	Name	Latency of a single operation (VLEN=1)
VADD	VDST ← VSRC1 + VSRC2	vector add	5 cycles
VMUL	VDST ← VSRC1 * VSRC2	vector mult.	15 cycles
VSHR	VDST ← VSRC >> 1	vector shift	1 cycles
VLD	VDST ← mem[SRC]	vector load	20 cycles
VST	VSRC → mem[DST]	vector store	20 cycles

- All the vector instructions operate with a vector length specified by VLEN. The VLD instruction loads VLEN consecutive elements from the DST address specified by the value in the VDST register. The VST instruction stores VLEN elements from the VSRC register in consecutive addresses in memory, starting from the address specified in DST.
- Both processors have eight vector registers (VR0 to VR7) which can contain up to 16 elements, and eight scalar registers (R0 to R7). The entire vector register needs to be ready (i.e., populated with all VLEN elements) before any element of it can be used as part of another operation.
- The memory can sustain a throughput of one element per cycle. The memory consists of 16 banks that can be accessed independently. A single memory access can be initiated in each cycle. The memory can sustain 16 parallel accesses if they all go to different banks.

(a) [10 points] Which processor (array or vector processor) is more costly in terms of chip area? Explain.

Array processor

Explanation: An array processor requires 16 functional units for an operation whereas a vector processor requires only 1.

(b) [25 points] The following code takes 52 cycles to execute on the vector processor:

```
VADD VR2 ← VR1, VR0
VADD VR3 ← VR2, VR5
VMUL VR6 ← VR2, VR3
```

What is the VLEN of the instructions? Explain your answer.

VLEN: 10

Explanation: 5+(VLEN-1)+5+(VLEN-1)+15+(VLEN-1) = 52 ⇒ VLEN = 10

How long would the same code execute on an array processor with the same vector length?

25 cycles

Explanation: there are data dependencies among instructions ⇒ 5+5+15= 25 cycles

(c) [25 points] The following code takes 94 cycles to execute on the vector processor:

```
VLD  VR0 ← mem[R0]
VLD  VR1 ← mem[R1]
VADD VR2 ← VR1, VR0
VSHR VR2 ← VR2
VST  VR2 → mem[R2]
```

Assume that the elements loaded in VR0 are all placed in different banks, and that the elements loaded into VR1 are placed in the same banks as the elements in VR0. Similarly, the elements of VR2 are stored in different banks in memory. What is the VLEN of the instructions? Explain your answer.

VLEN: 8

Explanation: 20+20+(VLEN-1)+5+(VLEN-1)+1+(VLEN-1)+20+(VLEN-1) = 94.
⇒ VLEN = 8

(d) [30 points] We replace the memory with a new module whose characteristics are unknown. The following code (the same as that in (c)) takes 163 cycles to execute on the vector processor:

```
VLD  VR0 ← mem[R0]
VLD  VR1 ← mem[R1]
VADD VR2 ← VR1, VR0
VSHR VR2 ← VR2
VST  VR2 → mem[R2]
```

The VLEN of the instructions is 16. The elements loaded in VR0 are placed in consecutive banks, the elements loaded in VR1 are placed in consecutive banks, and the elements of VR2 are also stored in consecutive banks. What is the number of banks of the new memory module? Explain.

[Correction] The number of cycles should be 170 instead of 163. For grading this question the instructor took into account only the student's reasoning.

Number of banks: 8

Explanation: Assuming that the number of banks is power of two, 20*(16/banks)+ 20*(16/banks)+ (banks-1)+ 5+ (VLEN-1)+ 1+ (VLEN-1)+ 20*(16/banks)+ (banks-1) = 170 ⇒ banks=8