6 Vector Processing [70 points]

A vector processor implements the following ISA:

Opcode	Operands	Latency (cycles)	Description
LD	V_{STR}, #n	1	$V_{STR} \leftarrow$ n (V_{STR} = Vector Stride Register)
LD	V_{LEN}, #n	1	$V_{LEN} \leftarrow$ n (V_{LEN} = Vector Length Register)
LDM	V_i	1	$V_{MSK} \leftarrow LSB(V_i)$ (V_{MSK} = Vector Mask Register)
VLD	V_i, #Address	22, pipelined	$V_i \leftarrow Mem[Address]$
VST	V_i, #Address	22, pipelined	$Mem[Address] \leftarrow V_i$
VADD	V_i, V_j, V_k	4, pipelined	$V_i \leftarrow V_j + V_k$
VMUL	V_i, V_j, V_k	6, pipelined	$V_i \leftarrow V_j * V_k$
VNOT	V_i	4, pipelined	$V_i \leftarrow BitwiseNOT(V_i)$
VCMPZ	V_i, V_j	4, pipelined	if($V_j == 0$) $V_i \leftarrow$ 0xFFFF; else $V_i \leftarrow$ 0x0000

Assume the following:

- The processor has an in-order core.
- The size of a vector element is 2 bytes.
- Each vector register V_i contains V_{LEN} vector elements. The total number of vector registers is 8.
- LD and LDM are not pipelined. They execute in one single cycle.
- LDM moves the least-significant bit (LSB) of each vector element in a vector register V_i into the corresponding position in V_{MSK}. This instruction is executed in one single cycle for all vector elements.
- V_{STR} and V_{LEN} are 16-bit registers. V_{MSK} has V_{LEN} bits.
- V_{MSK} enables predicated execution. Assume a simple implementation in which all V_{LEN} operations are executed, but the result writeback is turned off according to V_{MSK} (0 means writeback is turned off). Assume instructions LDM and VNOT are not subject to the mask (i.e., the result writeback is turned on for every vector element).
- The main memory is byte addressable.
- The main memory has N banks. N is a power of two. Vector elements stored in consecutive memory addresses are interleaved between the memory banks. For instance, if a vector element at address A maps to bank B, a vector element at address $A + 2$ maps to bank $(B + 1)$ % N, where % is the modulo operator and N is the number of banks.
- There is one single memory port, which is used for reads and writes.
- The processor *does not* support chaining between vector functional units.

(a) [5 points] What should the minimum value of N be to avoid additional stalls when executing a single VLD or VST instruction, assuming a vector stride of 1? Explain.

$N = 32$.

Explanation:
32 banks are needed because the latency of VLD and VST is 22 cycles, and N is a power of two.

Consider the following piece of code:

```
for (i = 0; i < 64; i++){
    if (A[i] == 0)
        B[i] = C[i];
    else
        B[i] = C[i] * A[i];
}
```

(b) [10 points] Translate the code into assembly language with the minimum number of instructions by using the provided ISA. Assume $V_{LEN} = 64$. (Hint: `C` should be loaded only once, before evaluating the `if` statement.)

```
LD VLEN, 64       # Load Vector Length Register
LD VSTR, 1        # Load Vector Stride Register
VLD V1, A         # Read from array A
VLD V2, C         # Read from array C
VCMPZ V3, V1      # Compare V1 to 0
LDM V3            # Load Vector Mask Register
VST B, V2         # Write to array B
VNOT V3           # BitwiseNOT
LDM V3            # Load Vector Mask Register
VMUL V4, V2, V1   # Multiply
VST B, V4         # Write to array B
```

(c) [15 points] What is the total number of cycles needed to execute the program in part (b)?

395 cycles.

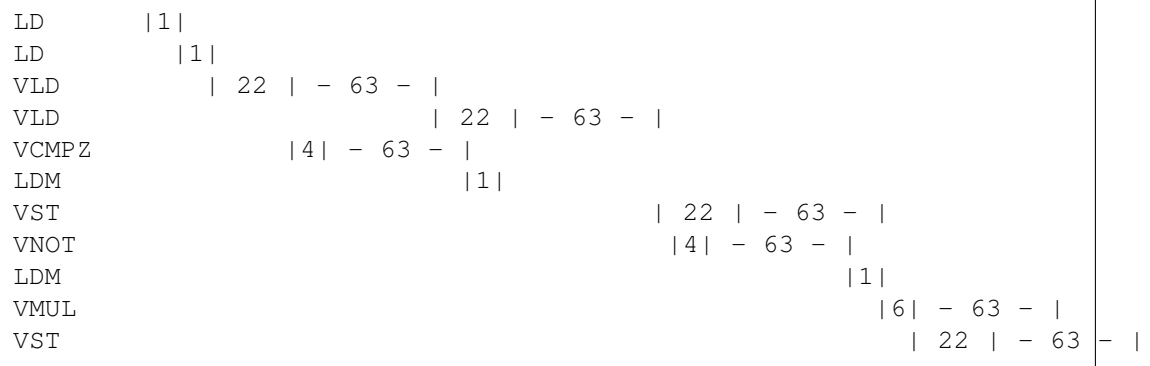
Explanation:

```
LD      |1|
LD        |1|
VLD         | 22 | - 63 - |
VLD                        | 22 | - 63 -  |
VCMPZ                       |4| - 63 - |
LDM                                  |1|
VST                                     | 22 | - 63 -    |
VNOT                                     |4| - 63 - |
LDM                                                 |1|
VMUL                                                  |6| - 63 - |
VST                                                                 | 22 | - 63 -|
```

(d) [15 points] If we execute the same code in a vector processor *supporting chaining*, what is the total number of cycles for the same program?

332 cycles.

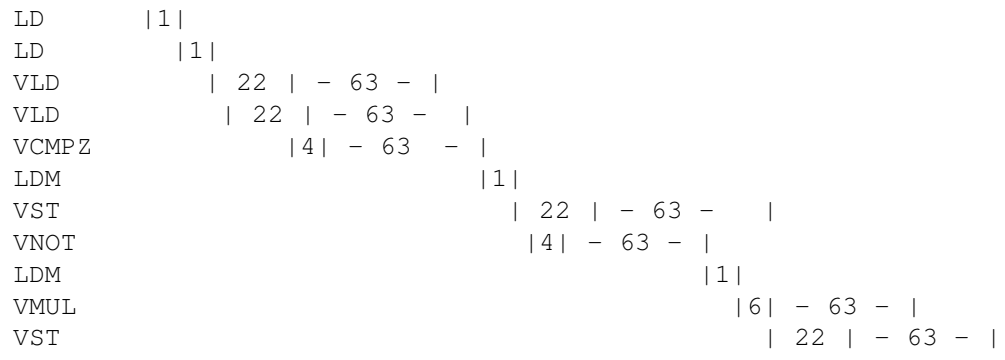
Explanation:



(e) [15 points] If we add one more memory port to the vector processor *supporting chaining*, what is the total number of cycles for the same program?

252 cycles.

Explanation:



(f) [10 points] A more efficient predicated execution is a density-time implementation, which scans the Vector Mask Register and only executes elements with non-zero masks. For the original vector processor (i.e., with one single memory port and no chaining), what would be the minimum number of cycles? (Note: Consider *no* overhead from scanning the Vector Mask Register.)

257 cycles.

Explanation:
The minimum number of cycles would be for all `A[i] = 0`, so that the last two instructions (`VMUL` and `VST`) wouldn't be executed.