Question 1:

1. After execution of 2 cycles, the values will be:

r[2] = 12, r[3] = 45, r[4] = 57, r[5] = -12

2. the code either tried to read new value by using sprn->r[3] or write to old value by using spro->r[1]. Both are wrong.

Question 2:

1. The first instruction takes 7 clock cycles, and the rest takes 6 clock cycles. So, in total 6n+1 clock cycles.
2. Yes, there are microarchitecture that can support memory access every clock cycle, for example pipelined microarchitecture.
3. Pros: simple, no hazards.

Cons: instructions cannot run in parallel, lower throughput.

Question 6:

We chose to implement the DMA USING threads. So that the copying and the main program could run in parallel.

DMA states:

REST – before any copy function was called, and after HALT.

WAIT – after the first copy function was called, but before the thread was prioritized by the OS.

We will also be at this state after finishing copying and before getting a new call for copy.

ACTIVE - once the thread was prioritized by the OS and starts copying, until the copy in done.

INSTRUCTIONS:

COPY – inputs are source address, destination address, and length. starts the copy operation.

POLL – Input is destination address. gets the number of bytes left to copy.