These are sample solutions, and there are other correct solutions too.

Question 1.

a) No additional components needed.

Fetch: \[ \text{IR} \leftarrow \text{Memory}[PC]; \]
\[ \text{PC} \leftarrow \text{PC} + 4; \]

Decode: \[ A \leftarrow \text{Reg}[\text{IR}[25:21]]; \]

Execution: \[ \text{ALUOut} \leftarrow A + \text{sign-extend}(\text{IR}[15-0]) \]

R-Type Completion: \[ \text{Reg}[\text{IR}[15:11]] \leftarrow \text{ALUOut} \]

b) No additional components needed.

Fetch: \[ \text{IR} \leftarrow \text{Memory}[PC]; \]
\[ \text{PC} \leftarrow \text{PC} + 4; \]

Decode: \[ A \leftarrow \text{Reg}[\text{IR}[25:21]]; \]
\[ B \leftarrow \text{Reg}[\text{IR}[20:16]] \]

- Assume these bits are all 0s

Execution: \[ \text{ALUOut} \leftarrow A + B \]

Completion: \[ \text{PC} \leftarrow \text{ALUOut} \]

c) A shift-left-16 component is needed for instruction[15-0] to multiplexer for ALU operand B.

Fetch: \[ \text{IR} \leftarrow \text{Memory}[PC]; \]
\[ \text{PC} \leftarrow \text{PC} + 4; \]

Decode: \[ A \leftarrow \text{Reg}[\text{IR}[25:21]]; \]

- Assume these bits are all 0s

Execution: \[ \text{ALUOut} \leftarrow A + \text{shift-left-16}(\text{IR}[15-0]) \]

Completion: \[ \text{Reg}[\text{IR}[20-16]] \leftarrow \text{ALUOut}; \]

Question 2.

Figure 1 shows the data path of such a processor.

Fetch: \[ \text{IR} \leftarrow \text{IMemory}[PC]; \]
\[ \text{PC} \leftarrow \text{PC} + 6; \]

Decode: \[ A \leftarrow \text{DMemory}[\text{IR}[47-32]]; \]
\[ B \leftarrow \text{DMemory}[\text{IR}[31-16]] \]

Execution: \[ \text{ALUOut} \leftarrow A - B \]

Completion: \[ \text{DMemory}[\text{IR}[31-16]] \leftarrow \text{ALUOut} \]

- If (ALUOut < 0) \[ \text{PC} \leftarrow \text{IR}[15-0]; \]

Figure 2 shows the state machine of this processor.
Figure 1: Datapath for 1-instruction processor

PCWrite = S0 + s3  MemRead = S1  MemWrite = S3  ALUOp = S2  IRWrite = S0

Question 3

add r3, r4, r2  F D X M W
sub r5, r3, r1  O O F D X M W
lw r6, 200(r3)  F D X M W
add r7, r3, r6

A change in the order can save one cycle

add r3, r4, r2  F D X M W
lw r6, 200(r3)  O O F D X M W
sub r5, r3, r1  F D X M W
add r7, r3, r6  O O F D X M W
Figure 2: state machine