

1. A.

[Answers below are examples, question is aimed at producing intelligent discussion]

Advantages

- 1) Having a small number of fixed formats simplifies control logic, for example source register addresses are always found in the same locations in the instructions and can be routed directly to the register address decoders.
- 2) Having a single instruction length means that we can make program jumps in instructions, rather than bytes, making the addressable range for jump instructions 4 times larger.
- 3) Single instruction length makes pipelining easier as we know we have a complete instruction on each 32 bit read.
- 4) We do not need to support shorter or longer instructions, all datapaths can be fixed at 32 bits wide.
- 5) We do not have to worry about instruction boundaries on caches.

Disadvantages

- 1) We cannot provide “arguments” to instructions in the following word or byte, for example an instruction to jump to the 32 bit address in the following word. We must load the address from memory into a register first and then do a jump register instruction.

B.

- i) In Immediate address mode, the operand is provided within the instruction itself, in the lower 16 bits. For example in an “add immediate” instruction, two registers are given, the source and destination and the value of the immediate data from bits 0 to 15 of the instruction are added to the contents of the source register and the result is placed in the destination register. The source register is not changed.
- ii) In Base + Index addressing the contents of one of the registers is regarded as the base address and the immediate data in the lower 16 bits of the instruction are the index. For example in a load word instruction the contents of the source register are added to the immediate data to get a memory address and the contents of this memory location are loaded into the destination register. The source register is not changed.

C.

The main datapath control unit generates the control signals for all of the components of the processor, based on the 6 bits of the opcode. The control signals determine which “routes” the data takes, for example whether the source of the second operand into the ALU is from a register or is immediate, and which operation the ALU performs. Example outputs are thus ALU control signals, and multiplexor control signals to manage routing.

2. A.

[hand drawn diagram to follow]

B.

[any similar discussion is acceptable]

Each of the two filter lights requires a separate output, but the “opposing” traffic lights can share outputs, for example while one set of lights is going through the sequence red / amber -> green -> amber the other set are always red, thus we could use three outputs (red, amber, green) to operate the sequence and a fourth to indicate which light should be running through the sequence and which should stay red. Therefore 6 outputs are required.

C.

Clearly the crossings would need buttons, which are additional inputs to the system and there are new states to cater for the period when all traffic lights are red but the pedestrian lights are green. There are also additional outputs required to control the pedestrian lights. There are design decisions to be made for example about whether traffic from all directions is stopped and the pedestrians allowed to cross, or whether we allow pedestrians across the “south” road while letting traffic flow east->west and west->east only. We also need to decide whether each crossing is independent (hence two additional inputs) or whether we combine both buttons into a single input and treat a button press as a request to cross both ways. These design decision will determine the additional inputs and states required.

3.

A.

In the form of product (“and”) terms involving all input variables that are then summed (“or”ed) together for “true” outputs. I.e. it is the set of input values that generate “true” outputs. It is a useful form as it lends itself to minimisation techniques such as Karnaugh maps.

B.

- i. A minterm is one of the product terms from a function expressed in canonical sum of products form.
- ii. An essential prime implicant is a minterm, or combination of minterms (e.g. Karnaugh Map group) that must form part of the final minimised function.
- iii. A redundant prime implicant is a minterm or combination of minterms that is already completely covered by one or more essential prime implicants.

C.

[Hand drawn diagram to follow]

D.

QMM is used in CPLD CAD to allow the designer to specify the required logic function in a form most appropriate to their needs, for example using a high level design language. QMM is then used internally by the CAD system to find the minimised function which can thus be implemented in the smallest

number of logic gates, thereby using as little of the CPLD as possible, or allowing the use of simpler CPLDs.

4.

A.

Exceptions are complicated to handle in a pipelined processor because instruction execution is overlapped, i.e. more than one instruction is being executed. Conceptually, exceptions happen in the “pause” between instructions but in a pipelined processor with overlapping instructions there is no such “pause”. When the exception is caused by a particular instruction (for example an add operation generates an overflow) then there is no real option other than to suspend execution of the pipeline; instructions already in the pipeline must be flushed and the results, if any discarded. If the exception occurs because of an external interrupt such as an indication from an external I/O device then there are more options for handling. We could stop after the current instruction and flush the pipeline as above; we could not load any new instructions and wait for the pipeline to empty, or we could wait for a particular type of instruction and interrupt that one.

B.

Throughput is the number of instructions executed in a given time. In a pipelined processor each instruction still takes the same elapsed time to execute, but various stages of the instruction are overlapped with other instructions. For example, if instructions take 5 clocks to execute in a multi-cycle processor then one instruction is started and completed every 5 clock ticks. In a pipelined processor with 5 stages at each clock tick one instruction completes and the 4th next instruction starts, thus the number of instructions completing in a given time is five times greater.

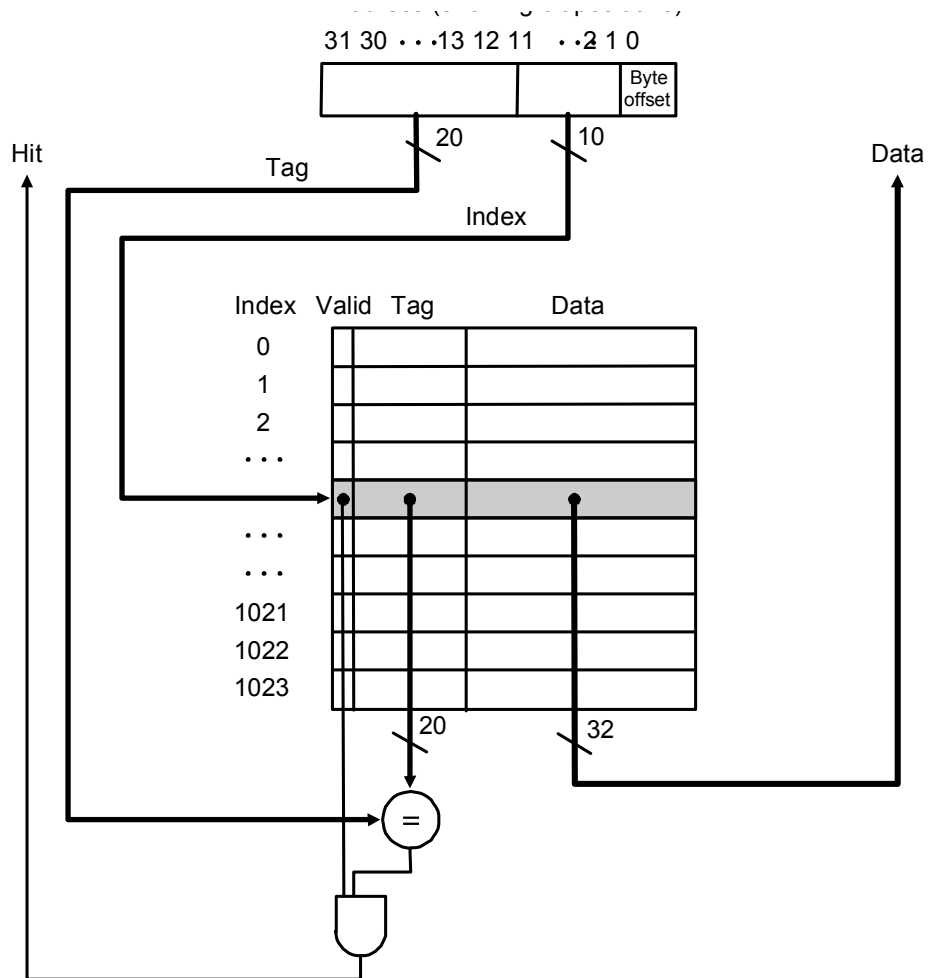
Longer pipelines theoretically give a greater speed up in throughput, however the longer the pipeline the more likely it is that hazards will occur as there are more instructions being overlapped. Hazards reduce pipeline performance as they may require bubbles to be inserted or the pipeline flushed, reducing the number of instructions completing. In particular, control hazards caused by branches may have greater penalties as there may be more instructions to “undo” if a branch is taken. Also, every pipeline stage requires additional control logic and buffering of both control signals and data.

C.

- i. A cache hit is finding the required data from a required address already located in the cache.
- ii. Hit time is the time required to search the cache, including retrieving the data.
- iii. The Miss Penalty is the time required to access the next level of cache, search for the required address and load the data into the current cache.

D.

[anything similar to below, or equivalent narrative description]



5.

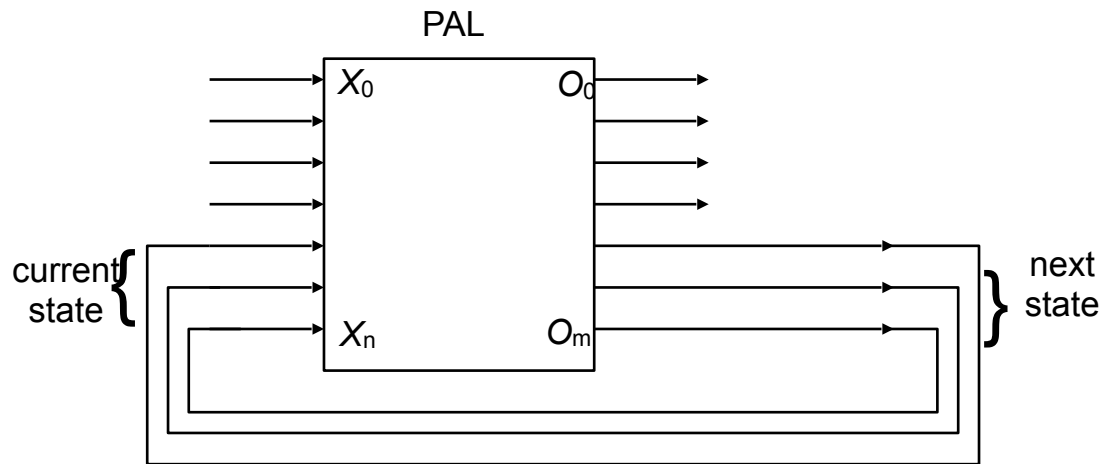
A.

A PLA has a series of inputs, which are available in their normal form and inverted. These are fed to an array of AND gates which can be selectively connected to any of the inputs or their inverses. The outputs of the AND gates are fed to an array of OR gates and each OR gate can be selectively connected to any of the AND gate outputs. Results of the OR gates are then fed (possibly buffered) to the PLA outputs. A PAL differs from this in having fixed connections to its OR plane.

[equivalent diagrams also acceptable]

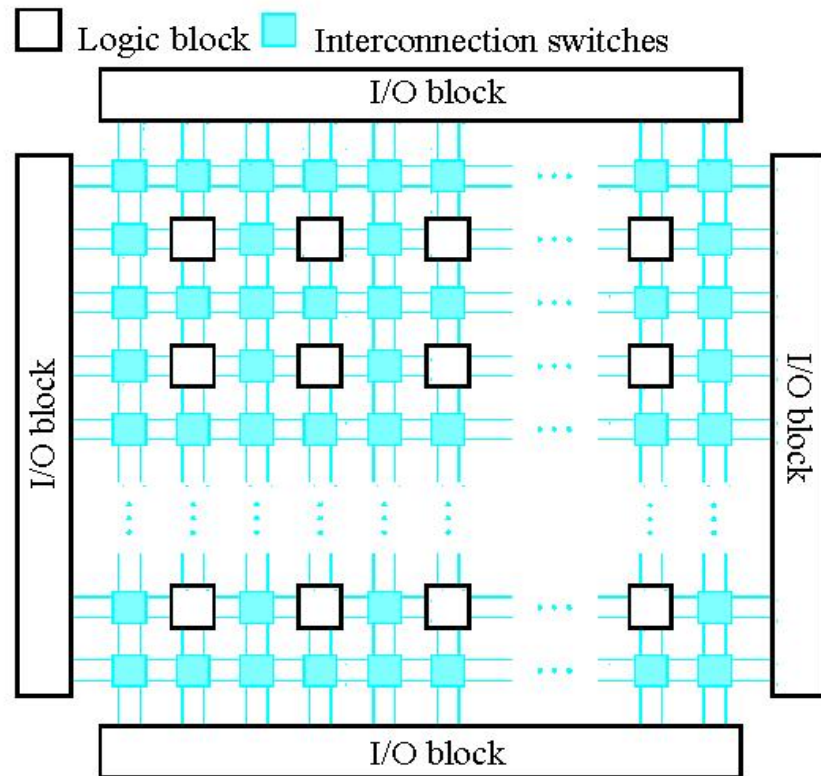
B.

(See below) A PAL is better than a ROM for state machine implementation because PALs have a flexible number of inputs and outputs while these are fixed in ROMs. ROMs require every state to be catered for, PALs allow us to minimise logic to take advantage of “don’t care” states and PALs are buffered and so do not require an external register.



C.

IO blocks provide buffered outputs and protected inputs to external devices. Logic blocks are similar to PLAs in that they provide a programmable array of logic devices that can be used to implement logic functions. Interconnection switches route data between the logic blocks and to IO blocks.



FPGAs can be used to implement complex devices such as controllers, interface cards or custom CPUs.

A.

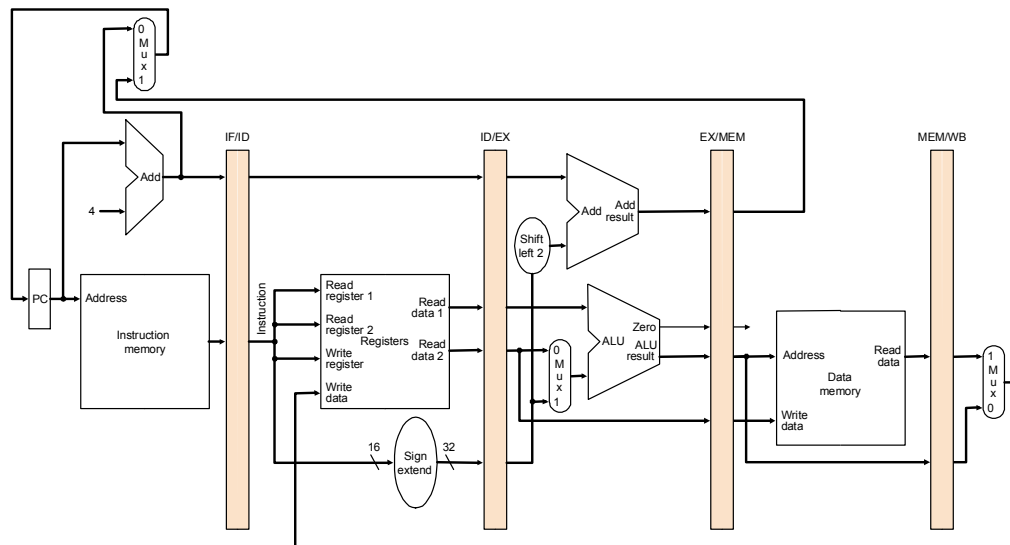
A single cycle processor suffers from the problem that the clock cycle must be long enough to allow the longest instruction path to be executed (this will usually be a load from memory operation). Since this is a single clock cycle processor all instructions will share the same clock length, that of the longest. Many instructions, for example register to register arithmetic can execute more quickly.

In moving to a multi-cycle design we allow instructions with short data paths to execute more quickly (i.e. in a small number of shorter clock cycles). We can now also share instruction and data memory as reads and write from this memory can occur in separate clock cycles, and we can also re-use components such as the ALU to both carry out arithmetic or logical instructions and calculate branch addresses for example. The multi-cycle approach however does require extra hardware as registers are needed to buffer the state of data between clock cycles and more control signals are needed with more complex control logic.

B.

A pipelined data is one in which the instruction processing has been split into a series of steps. The execution of steps can then be overlapped, with the second step of one instruction being executed at the same time as the first step of the next instruction. The number of possible overlapping steps is the length of the pipeline. A pipelined datapath is similar to an ordinary datapath but it will require registers between the hardware that implements each step and will need additional control logic. An example block diagram is shown below:

[anything bringing out the main features of the diagram below is acceptable]



C.

5.b. A pipeline hazard is a situation in which instructions cannot be overlapped, for example because the result of one instructions are used immediately by the next instruction. The next instruction cannot be overlapped because the result of the first instruction will not yet be available.

Control hazards are those in which the current instruction is a conditional branch. It may not be possible to overlap the execution the following instructions if the branch is taken. In the case of a conditional branch instruction for example, by the time the processor knows whether the branch will be taken the next instruction after the branch will already be loaded. The processor might as well execute this instruction. It is up to the compiler to make sure that this is either a useful instruction or is a no-operation. This is known as the *branch delay slot*. It only partly addresses the problem as only about 50% of the slot instructions will be needed.