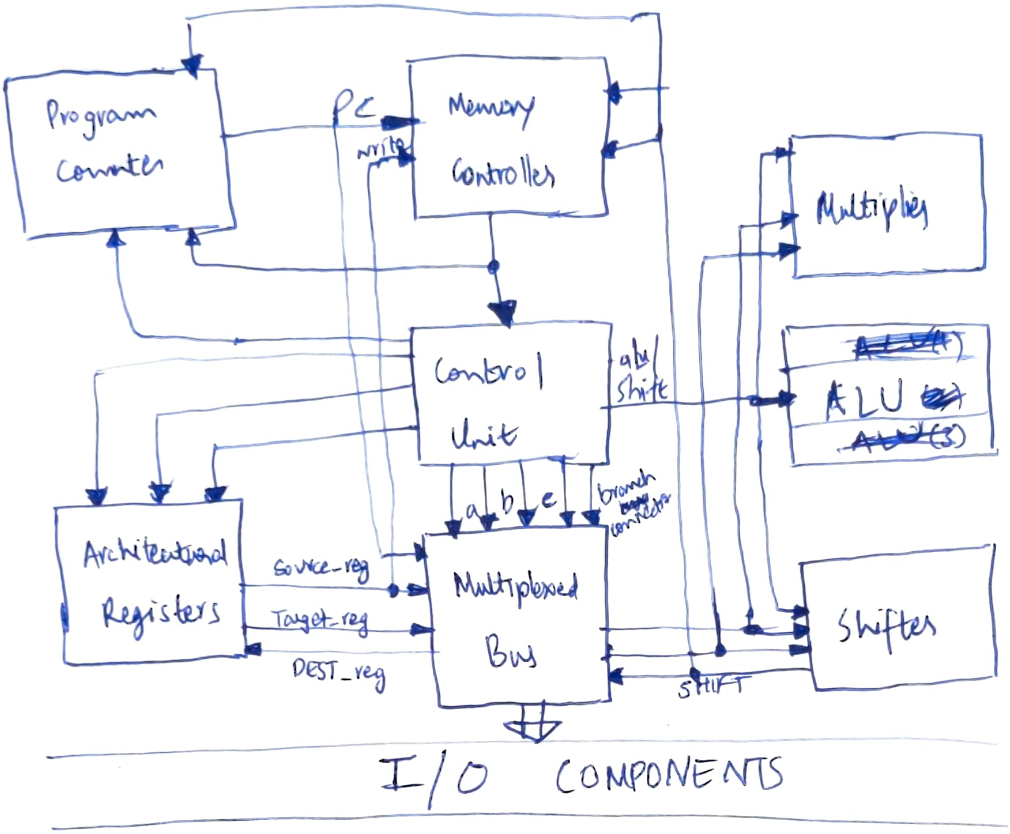
# COMPUTER ARCHITECTURE
# END-SEMESTER
# ASSESSMENT

Sreepathy Jayanand
CED17I038

1) a)



b) Pipelining is done and ALU is split to different functional units for out-of-order execution, in case of super-scalar. If we pipeline even the I/O components we can achieve better improvement in operation time.

c) Super-Scalar:

- CPU takes multiple instruction in the pipeline
- Executes several instructions at the same time.
- Ideal case : N instructions present in the processor
- Cost : Higher

This Design : (Super-pipelined)

- Breaks pipe line to give better efficiency per pipeline stage
- Issue rate = 1 in best and worst case
- Cost : Lesser compared to superscalar.

d) The superscalar will give higher performance because all the instructions are done ~~out ord~~ out-of-order compared to in-order exec$^n$ of our design.

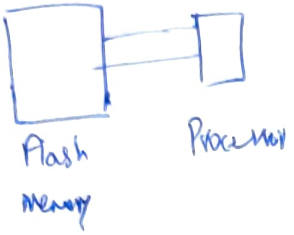
e) Inside the memory the OS would be loaded when the processor boots up from the hard-storage elements. Inside another segment program memory is loaded such that program memory only has read permissions on operating system but OS has RWE permissions.

f) DDR5 Memory: It is the latest and fastest and hence is better compared to previous generations.

g) A fully associative cache. Even though hit-time is increased it compensated by lower miss-rate in most-cases.

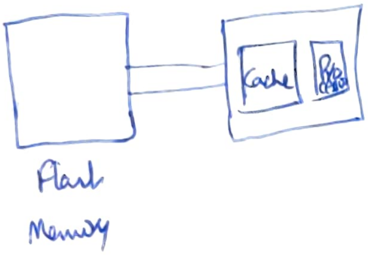
2. The per-bit cost of flash memory is less compared to that of cache memory. The speed of execution of cache memory is faster compared to flash memory.

a)



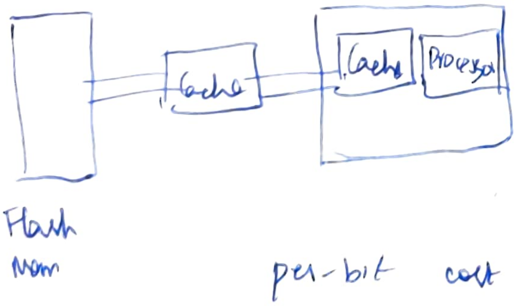
The data is taken from the flash memory by the processor. Since we use only flash memory the exec$^{n}$ is slowest and per-bit cost is lowest.

b)



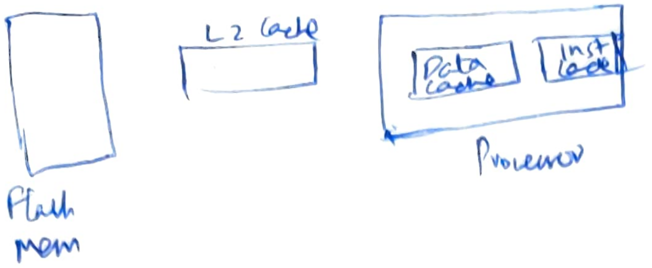
Processor with on-chip cache memory and then connected to flash memory device as storage will have the per-bit cost higher than that of (a) and the speed of execution being faster than (a) since the processor need not go till flash memory for the data.

c)



Processor with on-chip cache and off-chip cache and then connected to flash will have per-bit cost higher than (b) and speed of execution higher than (b) because of the hierarchical model followed. This is so because we go lesser times to the bottleneck (flash memory) compared to (b).

d)



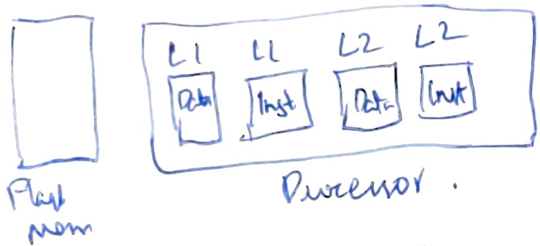
Processor with on-chip L1 cache containing data and instruction cache followed by L2 cache and flash memory.

Per bit cost > (c), because of additional hardware.

Speed of exec$^n$ > (c), because again we go lesser times to the slower flash memory.

Data cache takes care of read & write & instruction cache takes care of only read.

e)

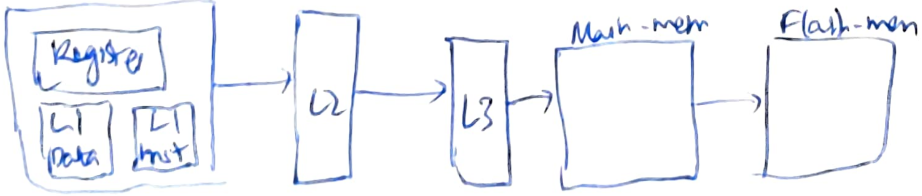


Has L1, L2 for both data and instruction cache.

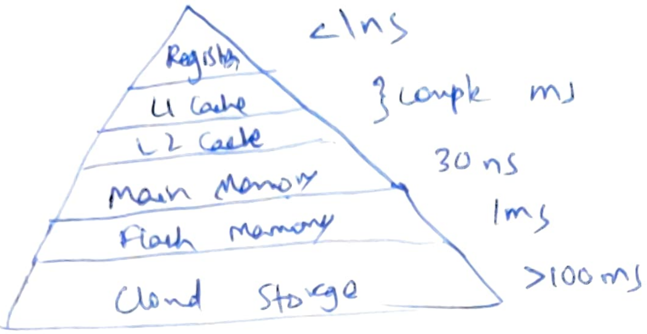
Per bit cost > (d), because of additional hardware.

Speed of exec$^n$ > (d), because more caches are present we need to access flash memory less times.

(e) performs the best compared to a, b, c, d, ~~e~~ but adding a register before the L1 cache will make it faster because of the more optimum hierarchy.



2) Cost & Time calculation.



a) speed ~ 1ms

cost ~ $4-12 per GB = $\frac{12}{8\times10^9}$ = $1.5\times10^{-9}$ $/bit

cost of

SRAM cache - ~ $5000 per GB

DRAM memory - ~ $20 per GB

Flash memory — ~ 4-12 per GB

Hard disk — $ 0.02 - 2.0 per GB

b) cost of on-chip cache memory (SRAM) ≃ $5000 per GB

≃ $\frac{5000}{10^9\times8}$ ≃ $62.5\times10^{-9}$ $/bit.

cost of flash memory ≃ $12 per GB

≃ $\frac{12}{10^9\times8}$ = $ $1.5\times10^{-9}$ $/bit.

Speed of exec^n of on-chip cache memory < 1ns

Speed of exec^n of flash memory ≃ 1ms.

Worst case time ≃ 1ms.

c) Cost of on-chip cache (SRAM) :- $5000 / GB , $6.25 \times 10^{-19}$ $ / bit.

Cost of off-chip cache as flash memory ≏ $12 / GB , $1.5 \times 10^{-9}$ $/bit

Cost of flash memory ≃ $12 / GB ~ $1.5 \times 10^{-9}$ $ / bit.

Speed of exec^n of on-chip cache ≃ <1ns.

Speed of exec^n of off-chip cache ≃ 10 ns.

Speed of exec^n of flash memory ≃ 1ms.

Worst case time ≃ 1ms.

d) Cost of on chip L1-cache ≃ $5000 / GB , $6.25 \times 10^{-19}$ / bit.

cost of on chip L2 Bus cache ≃ $5000 / GB , $62.5 \times 10^{-9}$ $/bit

cost of flash memory ≃ $12 / GB , $1.5 \times 10^{-9}$ $/bit

Speed of exec^n of L1 cache (both Instruction & Data) : <1ns

Speed of exec^n of L2 cache : ≃ 10 ns

Speed of exec^n of flash : ≃ 1ms

Worst case ≃ 1ms.

e) Cost of on chip L1 ≏ $62.5 \times 10^{-9}$ $ / GB

cost of on chip L2 cache ≈ $62.5 \times 10^{-9}$ $ / GB

Speed of exec^n of L1 cache : <1ns

Speed of exec^n of L2 cache : <1ns

Speed of exec^n of flash memory : 1ms

Worst case time ≈ 1ms

∴ option (d) is fastest.

3) a) Type B will have higher speed because in VLIW instructions are compiled by compiler based on independency and is lesser hardware intensive and more software based.

b) Limitations

A Type :

* Complex hardware
* Hardware dependent ILP
* Expensive

B Type :

* Compiler complexity is high
* If dependent instructions come NOP should be issued.
* memory system is complex

c) Multi-threaded support:

→ If a thread gets lots of cache misses the other threads can take advantage of resources

→ Lead to faster execution

→ Multiple threads can interface each other when sharing hardware.

d) VLIW

* Executed parallels
* Not complex hardware
* Compiler oriented

OOO

* Out of order exection
* Hardware oriented
* Simple software.

e) Multiple cores can help in multithreading using all cores • Help in multiple issue and out-of-order exec$^n$

Problems

Type A ~~Type 2B)~~

* Limited by
  - → # functional units
  - → # buses
  - → register ports

Type B

* Limited by
  - → code size
  - → no hazard detection (software)
  - → Dependency check using hardware

4) a) when the compiler is generating machine code and there are more live variables than number of registers and has to transfer or spill some variables the register spill problem occurs.

Can be overcome using:

* Increasing number of registers. (hardware solution)
* Avoiding loop unrolling
* Better scheduling.
* Storing excess variables in other registers temporarily

b) The limitation of ILP is register renaming.

Solution: Divide instructions to which need renaming and don't require renaming.

Instructions that don't require renaming is placed in additional memory.

c) Assume processor A commits 2 [illegible] instructions/cycle

B commits 4/cycle.

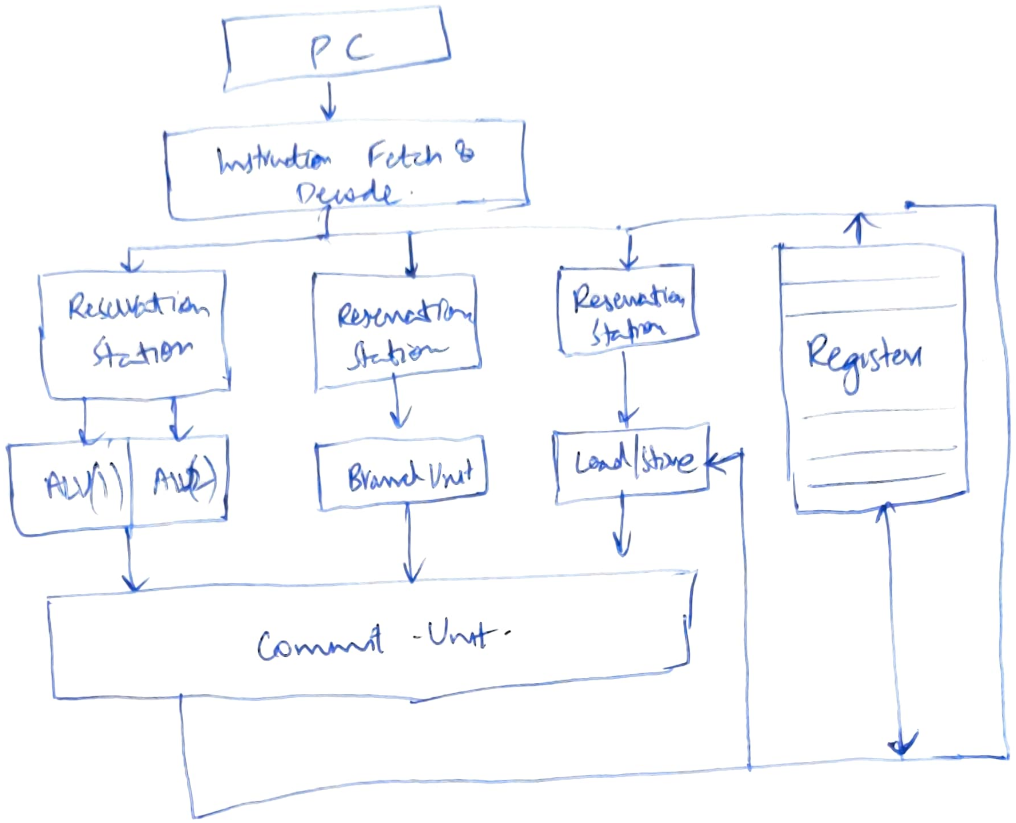
B perform 2x as of A.

If B performs 8/cycle, it is 4 times as good as A.

d) Using wallace tree multiplier we can perform integer multiplication as it already has SDRAM cells, shift, load ~~and~~, store and adders. It is hardware alone dependant and thus the company can successfully design the processor.

5)

a)



b) From the above architecture we can see that multiple units are connected to the commit unit. Hence by binning them into multiple commit unit we can go for multiple instruction issue and multiple commits on top of it.

c) More the number of cores, more instructions can be processed and hence more threads.