COMPUTER ARCHITECTURE
2020 - 2021

TD n°7 - Correction

Exercise 1: little and big-endian formats

The contents of a memory area and the value of certain data from that memory area are given.

Indicate whether the data is in big-endian, little-endian, or illegal format.

	0	1	2	3	4	5	6	7
123A0000	93	A4	77	1E	FB	86	79	83
123A0008	76	30	DE	BD	D9	78	CF	2F
123A0010	D6	44	94	4E	D4	4C	EA	02
123A0018	29	FA	87	2D	8A	EB	59	5B
123A0020	A8	65	3F	40	3A	0D	48	EE
123A0028	21	BA	DB	9B	1B	89	6F	47
123A0030	62	D6	80	A8	D1	9D	50	EE
123A0038	80	9B	D2	CB	D9	6B	5C	21

Addresses	Data
a) 123A0010	D644944E
b) 123A001A	2D87
c) 123A0031	D680 A8D1
d) 123A0020	EE480D3A403F65A8
e) 123A002C	1B896F47
f) 123A002B	9B1B896F
g) 123A001E	5B59
h) 123A0009	30
i) 123A000B	BDD9

a) 123A0010	D6 44 94 4E	big-endian
b) 123A001A	2D 87	little-endian
c) 123A0031	D6 80 A8 D1	illegal
d) 123A0020	EE 48 0D 3A 40 3F 65 A8	little-endian
e) 123A002C	1B 89 6F 47	big-endian
f) 123A002B	9B 1B 89 6F	illegal
g) 123A001E	5B 59	little-endian
h) 123A0009	30	little/big-endian
i) 123A000B	BD D9	illegal

Exercise 2: performance of input-output operations

Two devices are considered: a mouse and a hard disk drive.
Both peripherals are managed by programmed input/output.

An interrogation operation of a peripheral costs 100 clock cycles. The processor clock operates at a frequency of 50 MHz.

- The mouse must be interrogated 30 times per second to ensure that no user movement is missed.
- The disk transfers data to the processor in blocks of four bytes and has a throughput of 2 MB/second.

Calculate the fraction of processor time consumed in managing each of the devices.

Fraction of processor time consumed:

- The case of the mouse: it takes $30 \times 100 = 3000$ cycles per second to interrogate the mouse, which is $3\,000 / (50 \times 10^6) = 0{,}006$ % of processor cycles.

- The case of the hard disk drive: The disk transfers data at a rate of 2 MB/s. As the transfers are done in blocks of 4 bytes, 500 KB of queries per second are required.
  These 500 KB queries per second represent:
  $$500 \times 2^{10} \times 100 = 51{,}2 \times 10^6 \text{ cycles per second,}$$
  which constitutes $51{,}2 \times 10^6 / (50 \times 10^6) = 100$ % of processor cycles!

Exercise 3: performance of input-output operations

We now consider that the disk drive is managed by DMA:

- DMA initialisation by the processor requires 1000 clock cycles,
- Processing the interrupt at the end of the DMA transfer requires 500 clock cycles,
- Each DMA transfer involves 4 KB of data and the disk is 100% active.

What fraction of processor time is consumed by managing the DMA unit?

Each DMA transfer takes 4 KB/2 MB = 2×10^{-3} seconds.

As the disk transfers continuously, it needs:

$$(1\,000 + 500) / 2 \times 10^{-3} = 750 \times 10^{3} \text{ cycles per second,}$$

which constitutes $750 \times 10^{3} / (50 \times 10^{6}) = 1{,}5\ \%$ of processor cycles.

Exercise 4: polling and interruption

An I/O device makes ten requests per second, each of which requires five thousand processor instructions to process.

1) I/O is done by interrupt. It takes a thousand processor instructions to save the context and start the request handler, and another thousand processor instructions to load the context and return to the main process.
   How many instructions per second are needed to handle the I/O?

   Each request requires context saving, context processing and context loading, i.e., a total of 1000 + 5000 + 1000 instructions for each request. That is seventy thousand instructions per second to process the ten requests from the device.

2) I/O is now done by polling. The processor takes advantage of a pre-existing periodic interrupt every hundredth of a second to scan the device to see if there is a request to process. There is therefore no additional context switching cost.
   Each query requires five hundred processor instructions, in addition to processing the query if it is present.
   How many instructions per second are needed to handle the I/O?

The processor performs one hundred queries of the device per second, of which ninety are unsuccessful and ten result in requests being processed. The unsuccessful queries use five hundred processor instructions, while the remaining ten use 500 + 5,000 instructions (query and processing).

This gives one hundred thousand instructions per second:

$$(90 \times 500) + (10 \times (500 + 5\,000)) = 100\,000$$

Exercise 5: interruption priorities

The following four types of external or internal interruptions are considered:

- External interruption from a device,
- Attempting to execute an illegal instruction,
- Thermal interrupt indicating processor overheating,
- System call.

Rank these interruptions in order of priority (which interruption can interrupt the processing of a lower priority one?).

The thermal interrupt is the lowest priority because it does not require immediate processing (at the nanosecond level). There is no reason to give it priority when it can always be processed (by slowing down the processor, for example) once the other interrupts have been processed.

Attempting to execute an illegal instruction is the highest priority because any interrupt processing function can be wrong and contain an illegal instruction. It must therefore be possible to interrupt its execution.

An external interrupt from a peripheral, probably an I/O termination, requires processing by the operating system and thus the possibility of making a system call internally. The latter therefore has a higher priority. But there is no risk in slightly delaying an external interrupt arriving during the execution of a system call, until the end of it.

The interruptions are therefore classified as follows in descending order of priority:

- Attempt to execute an illegal instruction,
- System call,
- External interruption from a device,
- Thermal interrupt indicating processor overheating.