CSEC 20th – Past Year Paper Solution 2016-2017 Sem1

CE/CZ1006 – Computer Organisation and Architecture

Question 1

1)

a)

R2	SR
0XFF1	0x004
0X000	0x002
0X932	0x004
0XE25	0x004
0X000	0x000

b)

i)

START POP R2	R2 = 0x70F	
BSSR 1	SR = 0x001, C flag sets	
LOOP RLC R1	R1 = 0x3FD	R1 = 0x7FB
ADD R0, #0xFFF	R0 = 0x001, C flag sets	R0 = 0x000
JNE LOOP	Jump to Loop	Continue
MIDWAY ADD R3, R2		R3 = 0x910, V, N flags sets
NEXT MOV PC, #0x201		
PSH R1		
SUB R3, R3		
FINISH RET		

Therefore, at label MIDWAY,

R0 = 0x000, R1 = 0x7FB, R2 = 0x70F, R3 = 0x910, SR = 0x00C, SP = 0xFF3

ii) The instruction at NEXT says that move the immediate value 0x201 to PC. Therefore, the next line of execution will be 0x201. We go to the memory address 0x201:

ADDRESS	CONTENT
0X201	0x43C
0X202	0x001
0X203	0XBFD
0X204	0X000
:	:

We convert the machine code into mnemonics, which will be:

ADD R3, #001 ; add R3 by 1

JMP 0xFD ; PC = PC – 3

After the execution of JMP 0xFD, PC should point to 0x204. However, the instruction says PC = PC – 3, which means PC is 0x201 after the execution of JMP. Therefore, there is a infinite loop between 0x201 and 0x204. In each loop,

R3	SR	PC
0x911	0x004 (N flag sets)	0x203 -> 0x204(before E stage) -> 0x201(after E stage)
0x912	0x004	0x203 -> 0x204 -> 0x201
:	:	:
0xFFF	0x004	0x203 -> 0x204 -> 0x201
0x000	0x003 (Z, V flags set)	0x203 -> 0x204 -> 0x201
0x001	0x000 (all flags reset)	0x203 -> 0x204 -> 0x201
:	:	:
0x7FF	0x000	0x203 -> 0x204 -> 0x201
0x800	0x00C (N, V flags set)	0x203 -> 0x204 -> 0x201
0x801	0x004	0x203 -> 0x204 -> 0x201
:	:	:
0x911	0x004	0x203 -> 0x204 -> 0x201

Question 2

2)

a)

(I1) PSH [0x100]

Push value in memory register 0x100 (N) to stack.

(I2) PSH 0x101

The address of memory variable Ans is 0x101. Push it into stack.

(I3) ADD SP, #2

We pass in two variables as parameters of subroutine. Therefore, we have to remove them by adding 2 to Stack Pointer.

(I4) MOV R0, [SP + 4]

See what is Stack at this point

SP->	Base - 4	R0
+1	Base - 3	R1
+2	Base - 2	Address of I3
+3	Base - 1	#0x101 (argument 2)
+4	Base	N (argument 1)

To fetch the value of N, we find the address of SP + 4, and find the content of the register.

(I5) MOV R1, [SP + 3]

Similarly, the address of Ans can be got by finding the content of register SP + 3.

(I6) RET

Return from SumN

b) Notable, we can find i) the base condition, ii) the n to n -1 condition, and combine them together. When N = 0 end condition.

Otherwise,

SumN(N) = N + (N – 1) + (N – 2) + … + 2 + 1

= N + ((N - 1) + (N – 2) + … + 2 + 1

= N + Sumn(N - 1)

Therefore, our algorithm is:

```
IF N == 0:
    Ans is saved, exit.
ELSE:
    SumN(N) = N + SumN(N – 1)
```

Write this algorithm in assembly code

	CMP R0, #0	If R0 == 0
	JEQ END	Exit
	ADD [R1], R0	Ans = Ans + N
	DEC R0	N = N - 1
	PSH R0	Parameter 1(N) = N – 1
	PSH R1	Parameter 2(&Ans) = 0x101
	CALL SumN	Call SumN(N – 1, 0x101)
	ADD SP, #2	Destroy local frame
END	POPM 3	Recover content

c)

LOOP	CMP [Var1], [Var2]	While (Var1 < Var2)
	JGE Exit	Exit if Var1 >= Var2
	NEG [Var2]	Var2 = -Var2
	ADD [Var2], [Var1]	Var2 = Var1 + Var2
	JMP LOOP	Jump back
Exit	:	

Question 3

3)

a) (1) SRAM

A translational lookaside buffer (TLB) is a memory cache that is used to reduce the time taken to access a user memory location. It is part of the chip's memory-management unit (MMU). The TLB stores the recent translations of virtual memory to physical memory and can be called address-translation cache. A TLB may reside between the CPU and the CPU cache, between CPU cache and the main memory or between the different levels of the multi-level cache. TLB requires very high-speed IO from CPU. Therefore, TLB should be implemented with a very fast memory. SRAM is a good choice.

(2) NAND Flash

Thumb drive need to be fast in R/W, light and non-volatile. Therefore, RAM should not be chosen. Magnetic HDD is too heavy and large. NAND flash is cheaper and faster in R/W compared with NOR flash and EEPROM.

(3) Magnetic HDD

Storage memory is secondary memory, requiring cheap and non-volatile storage device. Data center involves frequently data exchange, therefore, NOR flash and NAND flash are not suitable with certain W/R lifecycle. HDD is cheap and has nearly infinite W/R times, which is suitable for the storage memory.

b) Fig1 below is a simplified illustration of a SRAM cell. Data is stored in M1, M2, M3, M4 gated. Those gates formed a lock that the states would not be changed if electrical is applied and Write Enable is disabled. Pass registers (M5, M6) are used to control the read or write enable of this cell.

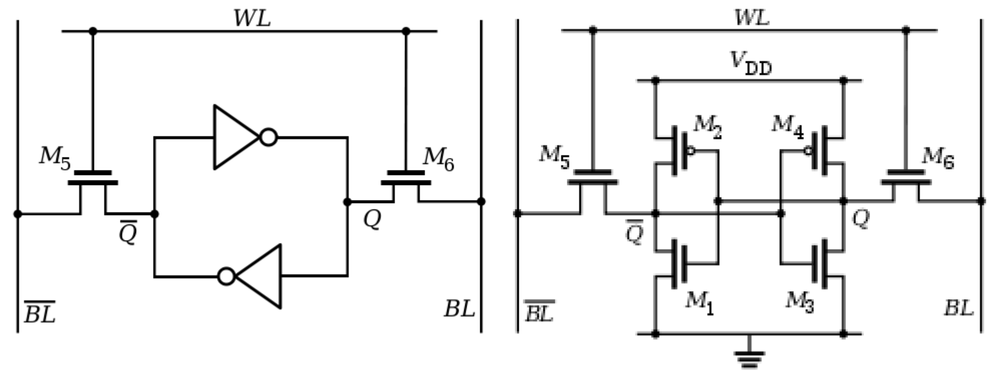

Fig1. Simplified illustration of a SRAM cell

c) 1. Eletrical signal level

The level of output voltage does not exceed the level of maximum input voltage, to prevent the input device from being short.

2. Communication protocol

The agreement or communication signal format should be the same between transmitter and receiver.

d)

i) Time for UART:

25 x 20µs = 500µs

Time for DMA:

500 µs	+	100 µs	+	50 µs x 2	+	100 µs = 800 µs
wait time for each request		transfer bus control from CPU to DMAC		transfer 1 word of data (assume 1 byte here) from UART Rx buffer to DMAC, and from DMAC to MEM		transfer bus control from DMAC to CPU

500 µs > 800 µs

The transfer time is determined by DMA, which is 800 µs. Number of packet per second 1 / 800

µs = 1250
And bits per packet = 1+ 8 + 1 = 10

Therefore, maximum baud rate = 1250 x 10 = 12500 bps, in between of 14400 bps and 9600 bps.
The maximum baud rate is 9600 bps.

ii) Transfer time for URAT: 500 µs
Transfer time for DMA in burst mode: 50 µs x 2 = 100 µs.
The time for transfer one package is determined by UART. Similarly, the maximum bits can be transferred per second is 1 / 500 µs x 10 = 20000 bps, in between of 19200 bps and 28800 bps. The maximum baud rate is 19200 bps.

iii) Absent from parity bit will lead to transferred data more prone to error. UART transfer is not able to detect an error, and may cause the use of wrong data.

Question 4

4)

a) Wearing levelling is a method used by SSD to write data in different blocks. SSD has limited number of erase cycles, thus writing data evenly throughout the whole disk can avoid a situation whereby a certain block is erased frequently. This extends the life of an SSD.

b) We can divide the problem into two parts:

Get the physical address and get the content of the certain address.

1. get physical address

EAT 1 = 80% x 10 (time of accessing TLB, hit) + 20% x (10 + 50) (time of accessing TLB, miss, accessing page table in main memory) = 20ns

2. get content

EAT 2 = 90% x 5 (time of accessing cache, hit) + 10% x (5 + 50) (time of accessing cache, miss, accessing main memory to copy the whole block) = 10ns Therefore, the total EAT = EAT 1 + EAT 2 = 30ns

c) 1. Add or subtract operands with the similar value of exponent first. They can produce fewer rounding errors.

2. Add or subtract operands before multiplying. Do division last. As multiplication has higher change to cause overflow, and division may cause the loss of precision.

d)

i)

1. Resource conflict

   Two instructions are trying to access the same resource in the same cycle.

2. Data conflict

   The value of source register of the next instruction is not ready at the execution stage, as the previous instruction is still updating its destination register (which is the same as the source register of the next instruction).

3. Branch:

   By the time of the finish of calculation of the target, unnecessary instruction is fetched and decoded.

ii)

Resource conflict I4, I7

I4 is storing a new result in [R0], while I7 is fetching [R0]. They are accessing memory using data bus at the same cycle.

Data conflict I3, I4

The content of register R1 will not be ready at the execution stage of I4, causing data conflict.

Branch 16, I7, 18

== End of Answers ==

Solver: Li Yuanming