CSA12 – Computer Architecture Analytical Answer Key

1. **Show the contents of registers E, A, Q, and SC during the process of division of (a) 10100011 by 1011; (b) 00001111 by 0011. (Use a dividend of eight bits.)**

Solution:

a)

$$\frac{10100011}{1011} = 1110 + \frac{1001}{1011} \qquad \frac{163}{11} = 14 + \frac{9}{11}$$

B = 1011 $\quad \overline{B} + 1 = 0101$ $\quad$ DVF = 0

	E	A	Q	SC
Dividend in AQ - - - -	0	1010	0011	100
shl EAQ - - - - -	1	0100	0110	
add $\overline{B} + 1$, suppress carry - -		0101		
$E = 1$, set Q_n to 1 - - - -	1	1001	0111	011
shl EAQ - - - - - -	1	0010	1110	
add $\overline{B} + 1$, suppress carry -		0101		
$E = 1$, set Q_n to 1 - - - -	1	0111	1111	010
shl EAQ - - - - - -	0	1111	1110	
add $\overline{B} + 1$, carry to E - -		0101		
$E = 1$, set Q_n to 1 - - - -	1	0100	1111	001
shl EAQ - - - - - -	0	1001	1110	
add $\overline{B} + 1$, carry to E - - -		0101		
$E = 0$, leave $Q_n = 0$ - - - -	0	1110	1110	
add B - - - - - - -		1011		
restore remainder - -	1	1001 (remainder)	1110 (quotient)	000

b)

$$\frac{1111}{0011} = 0101$$

B = 0011 $\quad \overline{B} + 1 = 1101$

	E	A	Q	SC
Dividend in Q, A = 0 - - - -		0000	1111	100
shl EAQ - - - - - -	0	0001	1110	
add $\overline{B} + 1$ - - - -		1101		
E = 0, leave $Q_n = 0$ - - - -	0	1110	1110	
add B - - - - - -		0011		
restore partial remainder - -	1	0001		011
shl EAQ - - - - - -	0	0011	1100	
add $\overline{B} + 1$ - - - - -		1101		
E = 1, set Q_n to 1 - - - - -	1	0000	1101	010
shl EAQ - - - - - - -	0	0001	1010	
add $\overline{B} + 1$ - - - - -		1101		
E = 0, leave $Q_n = 0$ - - - - -	0	1110	1010	
add B - - - - - -		0011		
restore partial remainder - - -	1	0001		001
shl EAQ - - - - - - - -	0	0011	0100	
add $\overline{B} + 1$ - - - - - -	1101			
E = 1, set Q_n to 1 - - - - -	1	0000 (remainder)	0101 (quotient)	000

2. Show the step-by-step multiplication process using Booth algorithm (as in Table 10-3) when the following binary numbers are multiplied. Assume 5-bit registers that hold signed numbers. The multiplicand in both cases is +15.

a. (+15) x (+13)

b. (+15) x (-13)

Solution:

a)

$(+15) \times (+13) = +195 = (0\ 011000011)_2$

BR = 01111 (+15); $\overline{BR} + 1 = 10001$ (–15); QR = 01101 (+13)

$Q_n\ Q_{n+1}$		AC	QR	Q_{n+1}	SC
	Initial	00000	01101	0	101
1 0	Subtract BR	10001			
		10001			
	ashr	11000	10110	1	100
0 1	Add BR	01111			
		00111			
	ashr	00011	11011	0	011
1 0	Subtract BR	10001			
		10100			
	ashr	11010	01101	1	010
1 1	ashr	11101	00110	1	001
0 1	Add BR	01111			
		01100			
	ashr	00110	00011	0	000
		+195			

b)

$(+15) \times (-13) = -195 = (1100\ 111101)_{2\text{'s comp,}}$

BR = 0 11111 (+15); $\overline{BR} + 1 = 10001$ (–15); QR = 10011 (–13)

Q_nQ_{n+1}		AC	QR	Q_{n+1}	SC
	Initial	00000	10011	0	101
1 0	Subtract BR	10001			
		10001			
	ashr	11000	11001	1	100
1 1	ashr	11100	01100	1	011
0 1	add BR	01111			
		01011			
	ashr	00101	10110	0	010
0 0	ashr	00010	11011	0	001
1 0	Subtract BR	10001			
		10011			
	ashr	11001	11101	1	000
		–195			

3. A DMA controller transfers 16-bit words to memory using cycle stealing. The words are assembled from a device that transmits characters at a rate of 2400 characters per second. The CPU is fetching and executing instructions at an average rate of 1 million instructions per second. By how much will the CPU be slowed down because of the DMA transfer?

Solution:

CPU refers to refers to memory on the average once (or more) 1 microsecond ($1/10^6$).

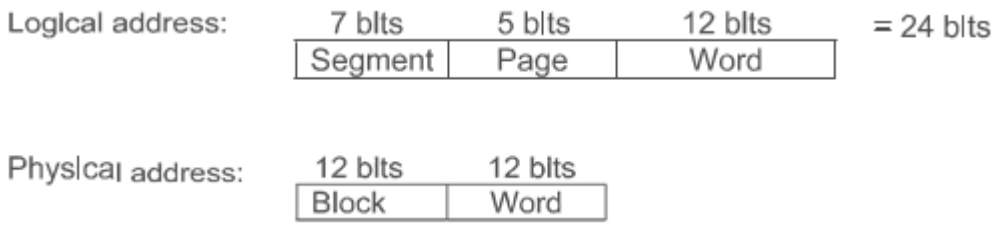
Characters arrive one every 1/2400=416.6 microsecond.

Two characters of 8 bits each are packed in to a 16-bit word every 2*416.6=833.3 microseconds.

The CPU is slowed down by no more than (1/833.3) *100=0.12%.

4. The logical address space in a computer system consists of 128 segments. Each segment can have up to 32 pages of 4K words in each. Physical memory consists of 4K blocks of 4K words in each. Formulate the logical and physical address formats.

Solution:



5. **A virtual memory system has an address space of 8K words, a memory space of 4K words, and page and block sizes of 1K words. The following page reference changes occur during a given time interval. (Only page changes are listed. If the same page is referenced again, it is not listed twice.)**

4 2012 6 1 4 0 1 0 2 3 5 7
Solution:

4 2 0 1 2 6 1 4 0 1 0 2 3 5 7

Page reference	(a) First-in		(b) LRU	
	Pages in main memory	Contents of FIFO	Pages in memory	Most recently used
Initial	0124	4201	0124	4201
2	0124	4201	0124	4012
6	0126	2016	0126	0126
1	0126	2016	0126	0261
4	0146	0164	1246	2614
0	0146	0164	0146	6140
1	0146	0164	0146	6401
0	0146	0164	0146	6410
2	1246	1642	0124	4102
3	2346	6423	0123	1023
5	2345	4235	0235	0235
7	2357	2357	2357	2357

6. **Convert the following binary numbers to decimal: 101110; 1110101; and 110110100.**

$(101110)_2 = 32 + 8 + 4 + 2 = 46$
$(1110101)_2 = 64 + 32 + 16 + 4 + 1 = 117$
$(110110100)_2 = 256 + 128 + 32 + 16 + 4 = 436$

7. **Perform the arithmetic operations (+70) + (+80) and (-70) + (-80) with binary numbers in signed-2's complement representation. Use eight bits to accommodate each number together with its sign. Show that overflow occurs in both cases, that the last two carries are unequal, and that there is a sign reversal.**

	01 ← last two carries → 10		
+ 70	01000110	– 70	10111010
+ 80	01010000	– 80	10110000
+150	10010110	– 150	01101010
↑	↑	↑	↑
greater than +127	negative	less than – 128	positive

8. **A computer uses a memory unit with, 256K< words of 32 bits each. A binary instruction code is stored in one word oi memory. The instruction has four parts: an indirect bit, an operation code, a register code part to specify one of 64 registers, and an address part. a. How many bits are there in the operation code, the register code part, and the address part? b. Draw the instruction word format and indicate the number of bits in each part. c. How many bits are there in the data and address inputs of the memory?**

$256\ K = 2^8 \times 2^{10} = 2^{18}$
$64 = 2^6$

(a) Address: 18 bits
Register code: 6 bits
Indirect bit: 1 bit
25 32 – 25 = 7 bits for opcode.

(b) 1 7 6 18 = 32 bits

I	opcode	Register	Address

(c) Data; 32 bits; address: 18 bits.

$256\ K = 2^8 \times 2^{10} = 2^{18}$

5 3 6 18 = 32

Address = 18 bits
Mode = 3 "
Register = 6 "
27 bits
op code 5
32 bits

9. **Write a program to multiply two positive numbers by a repeated addition method. For example, to multiply 5 X 4, the program evaluates the product by adding 5 four times, or 5 + 5 + 5 + 5.**

```
      LDA    A     /Load multiplier
      SZA          /Is it zero?
      BUN    NZR
      HLT          /A=0, product = 0 in AC
NZR,  CMA
      INC
      STA    CTR   /Store –A in counter
      CLA          /Start with AC = 0
LOP,  ADD    B     /Add multiplicand
      ISZ    CTR
      BUN    LOP   /Repeat Loop A times
      HLT
A,    DEC    –     /multiplier
B,    DEC    –     /multiplicand
CTR,  HEX    O     /counter
      END
```

10. **Consider the two 8-bit numbers A =01000001 and B =10000100.**
a. Give the decimal equivalent of each number assuming that (1) they are unsigned, and (2) they are signed.
b. Add the two binary numbers and interpret the sum assuming that the numbers are (1) unsigned, and (2) signed.
c. Determine the values of the C, Z, S, and V status bits after the addition.

	Unsigned	Signed
A = 01000001	65	+ 65
B = 10000100	132	– 124
A + B = 11000101	197	– 59

(c) C = 0 Z = 0 S = 1 V = 0
(d) BNC BNZ BM BNV

11. Obtain the l's and 2's complements of the following eight-digit binary numbers:

(a) 10101110;
(b) 10000001;
(c) 10000000;
(d) 00000001;
(e) 00000000.

Solution

1.

10101110

1's complement. 01010001

2's complement 01010010

2.

10000001

1's complement 01111110

2's complement 01111111

3.

10000000

1's complement. 01111111

2's complement . 10000000

4.

00000001

1's complement. 11111110

2'scomplement . 11111111

5.

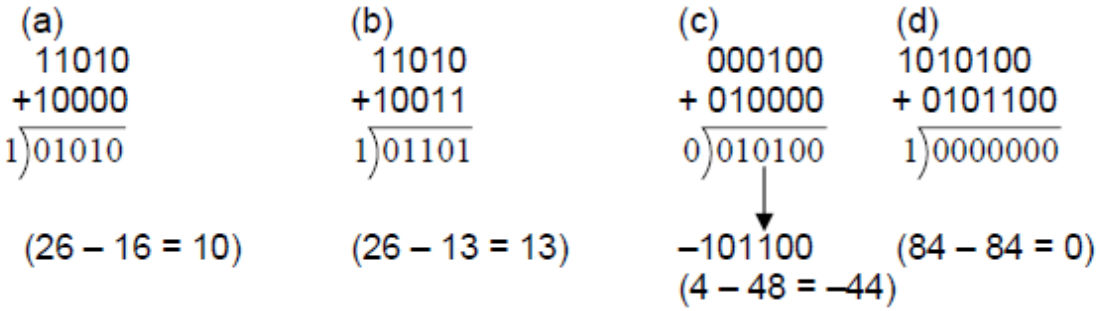
00000000

1's complement . 11111111

2's complement. 00000000

12. Perform the subtraction with the following unsigned binary numbers by taking the 2's complement of the subtrahend.

Solution



a. 11010 - 10000

First take 2's complement of 10000. 2's complement can be found out:

10000 → 01111 + 1 = 10000

M = 11010 ; N = 10000

Here, M>N

Add M to 2's complement of N.

M = 11010

2's complement of N = 10000

The sum is 101010

Answer : 01010

Hence, answer for (i) is **01010.**

Similarly, the answer for (ii) is **9909.**

The answer for (iii) is **10990.**

b. 11010 – 1101

Here A = 11010, B = 01101

First find 2's complement of B = 01101

2's complement = 1's complement + 1

1's complement of 01101 is 10010

Now add 1 : 10010 + 1 = 10011

Add 2's complement of B and A

11010 + 10011 = 101101

Since carry is generated. Ignore the leftmost bit.

So, the answer is 01101

C. 100 - 110000

Here A = 000100, B = 110000

First find 2's complement of B = 110000

2's complement = 1's complement + 1

1's complement of 110000 is 001111

Now add 1 : 001111 + 1 = 010000

Add 2's complement of B and A

000100 + 010000 = 010100

Since there is no carry generated, again find the 2's complement of 010100

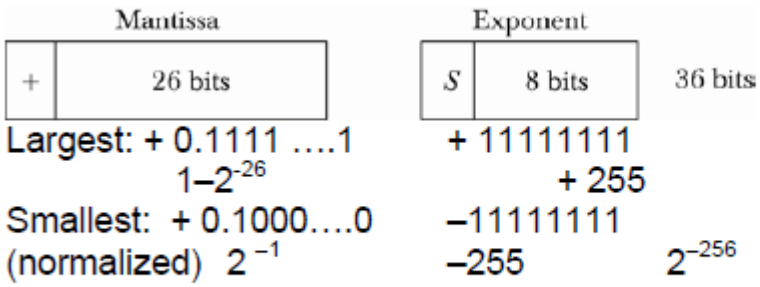
1's complement of 010100 is 101011

Now add 1 : 101011 + 1 = 101100

So, the answer is -101100

13. A 36-bit floating-point binary number has eight bits plus sign for the exponent and 26 bits plus sign for the mantissa. The mantissa is a normalized fraction. Numbers in the mantissa and exponent are in signed-magnitude representation. What are the largest and smallest positive quantities that can be represented, excluding zero?

Solution:



The range of number which it can accommodate for a 36-bit floating point binary number is as follows :-

2 bits reserved for sign

26 bits for mantissa

8 bits for exponent

So the range will be :-

$-(1-2^{-26}) \times 2^{255}$ **to** $(1-2^{-26}) \times 2^{255}$

Largest Quantity : $(1-2^{-26}) \times 2^{255}$

Smallest Quantity : $-(1-2^{-26}) \times 2^{255}$

14. A nonpipellne system takes 50 ns to process a task. The same task can be processed in a six-segment pipeline with a dock cycle of 10 ns. Determine the speedup ratio of the pipeline for 100 tasks. What is the maximum speedup that can be achieved?

Solution

$t_n = 50$ ns
$k = 6$
$t_p = 10$ ns
$n = 100$

$$S = \frac{nt_n}{(k+n-1)t_p} = \frac{100 \times 50}{(6-99) \times 10} = 4.76$$

$$S_{max} = \frac{t_n}{t_p} = \frac{50}{10} = 5$$

Concept:

Speed up factor is defined as the ratio of time required for non-pipelined execution to that of time received for pipelined execution.

Data:

Time for non-pipelined execution per task = tn = 50 ns
Time for pipelined execution per task = tp = 10 ns
Number of stages in the pipeline = k = 6
Number of tasks = 100

Formula

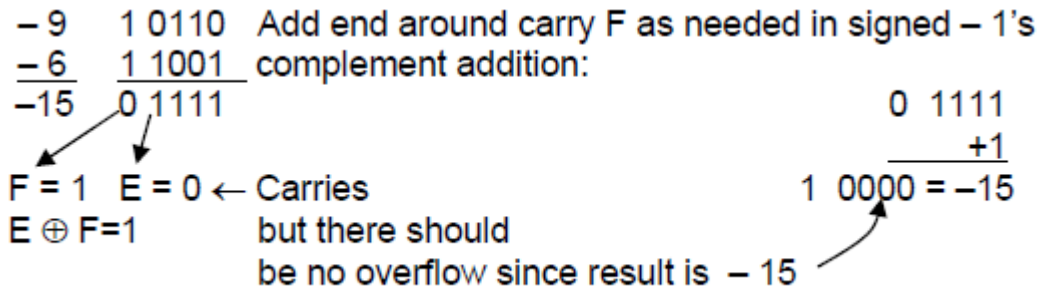
S = Tn/Tp
S = speed up factor

Calculation:
Time for non-pipelined = Tn = tn × Number of tasks
Time for non-pipelined = Tn = 50 × 100 = 5000
Time for pipelined = Tp = 1st task × k × tp + (All Remaining Tasks (k - 1)) × tp
Time for pipelined = Tp = 1 × 6 × 10 + (100 - 1) × 10 = 1050

S = Tn/Tp = 4.7

15. **Perform the operation (-9) + (- 6) = - 15 with binary numbers in signed-1' s complement representation using only five bits to represent each number (including the sign). Show that the overflow detection procedure of checking the inequality of the last two carries fails in this case.**

Solution



16. Convert the following decimal numbers to the bases indicated. a. 7562 to octal b. 1938 to hexadecimal c. 175 to binary

Solution

(7562)10 = (16612)8

(1938)10 = (792)16

(175)10 = (10101111)2

17. Perform the arithmetic operations (+42) + (- 13) and (-42) - (- 13) in binary using signed-2's complement representation for negative number

Solution

+ 42 = 0101010

– 42 = 1010110

+13 = 0001101

–13 = 1110011

(+42) 0101010	(– 42) 1010110
(–13) 1110011	(+ 13) 0001101
(+29) 0011101	(– 29) 1100011

18. Consider a computer with four floating-point pipeline processors. Suppose that each processor uses a cycle time of 40 ns. How long will it take to perform 400 floating-point operations? Is the difference lf the same400operatlons are carried out using a single pipeline processor with a cycle time of 10 ns?

Solution

Divide the 400 operations into each of the four Processors,

Processing time is: (400 /40)*4 =4,000 nsec.

Using a single pipeline, processing time is 400 to 4000 nsec.

19

a. Design and analysis of a space time diagram for a 6 segment pipeline showing the time it takes to process eight tasks.

b. Determine the number of clock cycles that it takes to process 200 tasks in a 6 segment pipeline.

Segment	1	2	3	4	5	6	7	8	9	10	11	12	13
1	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8					
2		T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8				
3			T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8			
4				T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8		
5					T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	
6						T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8

$(k + n - 1)t_p = 6 + 8 - 1 = 13$ cycles

k = 6 segments
n = 200 tasks $(k + n - 1) = 6 + 200 - 1 = 205$ cycles

20 **A computer employs RAM chips of 256 x 8 and ROM chips of 1024 x 8. The computer system needs 2K bytes of RAM, 4K bytes of ROM, and four interface units, each with four registers. A memory-mapped 1/0 configuration is used. The two highest-order bits of the address bus are assigned 00 for RAM, 01 for ROM, and 10 for interface registers. a. How many RAM and ROM chips are needed? b. Draw a memory-address map for the system. c. Give the address range in hexadecimal for RAM, ROM, and interface**

RAM	2048 /256 = 8 chips;	$2048 = 2^{11}$; $256 = 2^8$
ROM	4096 /1024 = 4 chips;	$4096 = 2^{12}$; $1024 = 2^{10}$
Interface	4 × 4 = 16 registers; $16 = 2^4$	

Component	Address	16 15 14 13 12 11 10 19	8765	4321
RAM	0000-O7FF	0 0 0 0 0 ← 3×8 decoder	xxxx	xxxx
ROM	4000-4FFF	0 1 0 0 ← 2×4 decoder	xx xxxx	xxxx
Interface	8000-800F	1 0 0 0 0 0 0 0	0000	xxxx