Assignment Project Exam Help

S

https://eduassistpro.github.io/

COMSC

Add WeChat edu_assist_pro

Assembly La

Computer Organization

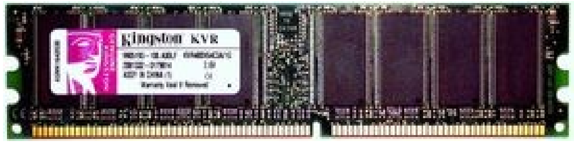
Outline

- Welcome to COMSC 260
- Assembly-, Machine-, and High-Level Languages
- Assembly Language Programming Tools
- Programmer’ System
- Basic Computer Organizati

Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

Memory Devices

- Random-Access Memory (RAM)
  - Usually called the main memory
  - It can be read and written to
  - It does not store information permanently (Volatile , when it is powered off, the stored information are gone)
  - Information stored in it can b e periods (hence the name random access)
  - Information is accessed by an address that specifie tion of the piece of information in the RAM.
  - DRAM = Dynamic RAM
    - 1-Transistor cell + trench capacitor
    - Dense but slow, must be refreshed
    - Typical choice for main memory
  - SRAM: Static RAM
    - 6-Transistor cell, faster but less dense than DRAM
    - Typical choice for cache memory



Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

Memory Devices

- ROM (Read-Only-Memory)
  - A read-only-memory, non-volatile i.e. stores information perm

Assignment Project Exam Help

  - Has random access of stored information
  - Used to store the inform computer

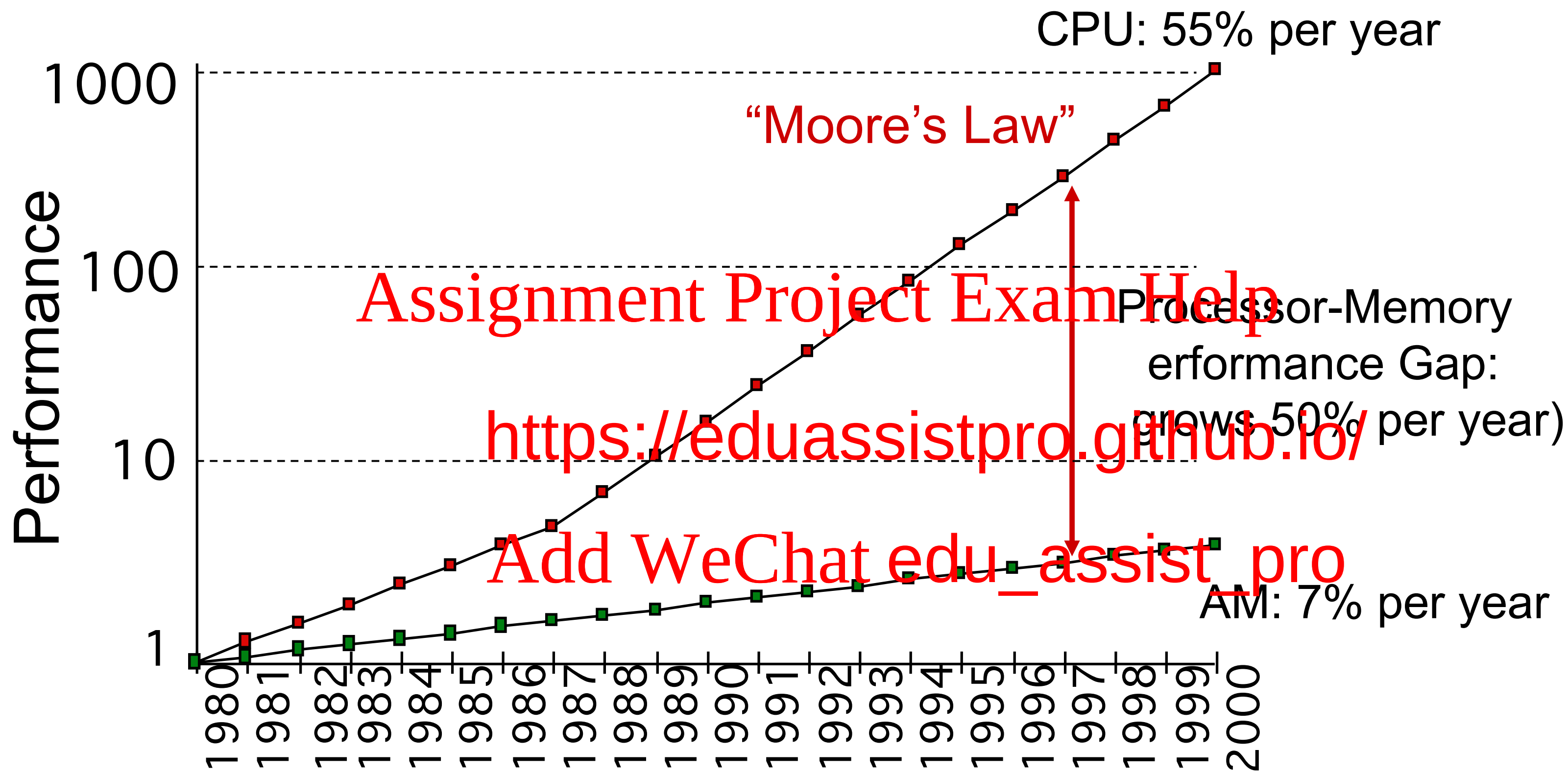
https://eduassistpro.github.io/

  - Many types: ROM, EPRO
  - FLASH memory can be erased electrically i

Add WeChat edu_assist_pro

- Cache
  - A very fast type of RAM that is used to store information that is most frequently or recently used by the computer
  - Recent computers have 2-levels or more levels of cache; the first level is faster but smaller in size (usually called internal cache), and the second level is slower but larger in size (external cache).

Processor-Memory Performance Gap



Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

- 1980 – No cache in microprocessor
- 1995 – Two-level cache on microprocessor

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

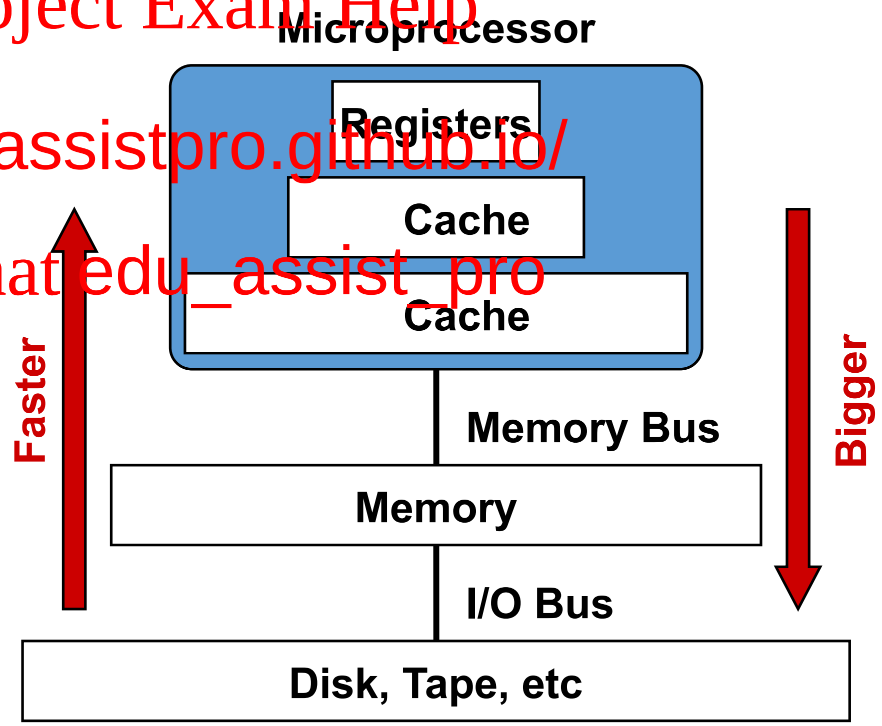
The Need for a Memory Hierarchy

- Widening speed gap between CPU and main memory
  - Processor operation takes less than 1 ns
  - Main memory requires more than 50 ns to access
- Each instructio memory access
  - One memory on
  - Additional memory accesses f s involving memory data access
- Memory bandwidth limits the instruction execution rate
- Cache memory can help bridge the CPU-memory gap
- Cache memory is small in size but fast

Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

Typical Memory Hierarchy

- Registers are at the top of the hierarchy
  - Typical size < 1 KB
  - Access time < 0.5 ns
- Level 1 Cache (8 – 64 KB)
  - Access time: 0.5 – 1 ns
- L2 Cache (512KB – 8MB)
  - Access time: 2 – 10 ns
- Main Memory (1 – 2 GB)
  - Access time: 50 – 70 ns
- Disk Storage (> 200 GB)
  - Access time: milliseconds



Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

Magnetic Disk Storage

Disk Access Time =

Seek Time +

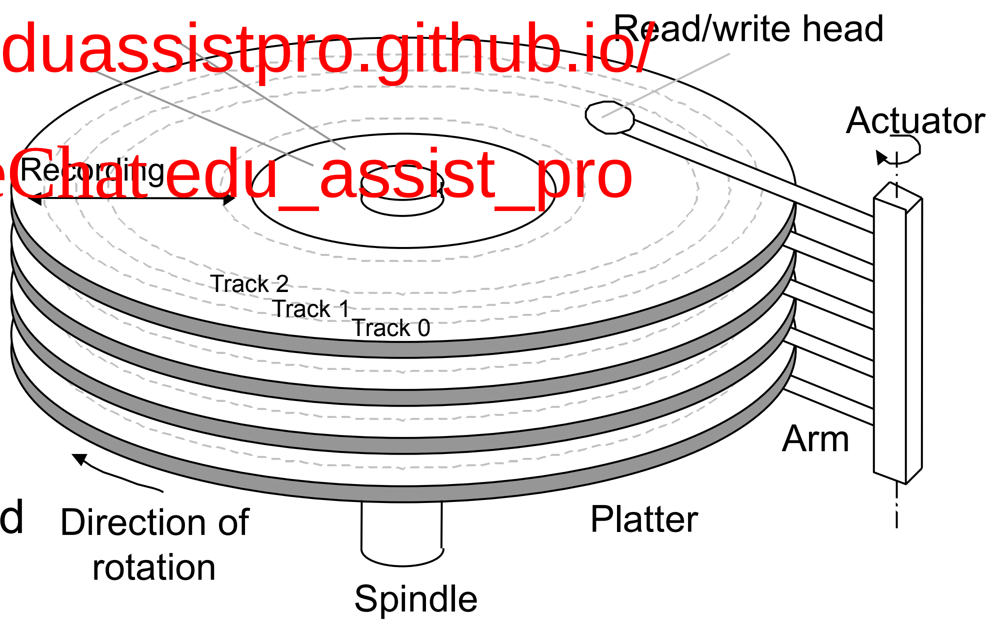
Rotation Latency +

Transfer Time

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro



Seek Time: head movement to the desired track (milliseconds)

Rotation Latency: disk rotation until desired sector arrives under the head

Transfer Time: to transfer data

Example on Disk Access Time

- Given a magnetic disk with the following properties
  - Rotation speed = 7200 RPM (rotations per minute)
  - Average seek = 8 ms, Sector = 512 bytes, Track = 200 sectors

Assignment Project Exam Help

- Calculate
  - Time of one r

https://eduassistpro.github.io/

  - Average time to access a bloc cutive sectors

Add WeChat edu_assist_pro

- Answer
  - Rotations per second = 7200/60 = 120 RPS
  - Rotation time in milliseconds = 1000/120 = 8.33 ms
  - Average rotational latency = time of half rotation = 4.17 ms
  - Time to transfer 32 sectors = (32/200) * 8.33 = 1.33 ms
  - Average access time = 8 + 4.17 + 1.33 = 13.5 ms

Assignment Project Exam Help

Da tion

https://eduassistpro.github.io/

Chapte

Add WeChat edu_assist_pro

Outline

- Introduction
- Numbering Systems
- Binary & Hexadecimal Numbers
- Base Conversions
- Integer Storage Sizes
- Binary and Hexadecimal Addition
- Signed Integers and 2's Complement Notation
- Binary and Hexadecimal subtraction
- Carry and Overflow
- Character Storage

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Introduction

- Computers only deal with binary data (0s and 1s), hence all data manipulated by computers must be represented in binary format.
- Machine instructions manipulate many different forms of data:
  - Numbers:
    - Integers: 33, +128, -2827
    - Real numbers: 1.33, +9.5560
  - Alphanumeric characters (letters, numbers, signs, characters): examples: A, a, c, 1 ,3, ", +, Ctrl, Shift, etc.
  - Images (still or moving): Usually represented by numbers representing the Red, Green and Blue (RGB) colors of each pixel in an image,
  - Sounds: Numbers representing sound amplitudes sampled at a certain rate (usually 20kHz).
- So in general we have two major data types that need to be represented in computers; numbers and characters.

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

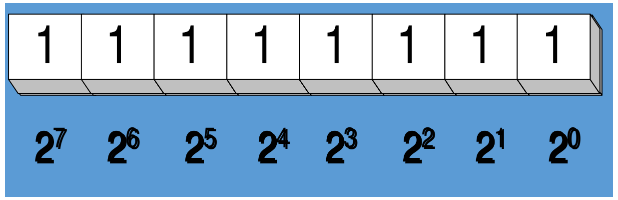
Numbering Systems

- Numbering systems are characterized by their base number.
- In general a numbering system with a base *r* will have *r* different digits (including the 0) in its its will range from 0 to *r*-1
- The most widely used numbering sy listed in the table below:

Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

Numbering System	Base	Digits Set
Binary	2	1 0
Octal	8	7 6 5 4 3 2 1 0
Decimal	10	9 8 7 6 5 4 3 2 1 0
Hexadecimal	16	F E D C B A 9 8 7 6 5 4 3 2 1 0

Binary Numbers



- Each digit (bit) is either 1 or 0
- Each bit represents a power of 2
- Every binary number i

Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

Converting Binary to Decimal

- Weighted positional notation shows how to calculate the decimal value of each binary bit:

Assignment Project Exam Help

$Decimal = (d_{n-1}$ [illegible] $+ (d_1 \times 2^1) + (d_0 \times 2^0)$

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

d = binary digit

- binary 10101001 = decimal 169:

$(1 \times 2^7) + (1 \times 2^5) + (1 \times 2^3) + (1 \times 2^0) = 128+32+8+1=169$

Convert Unsigned Decimal to Binary

- Repeatedly divide the decimal integer by 2. Each remainder is a binary digit in the translated value:

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

least significant bit

most significant bit

stop when quotient is zero

37 = 100101

Another Procedure for Converting from Decimal to Binary

- Start with a binary representation of all 0’s
- Determine the highest possible power of two that is less or equal to the number.
- Put a 1 in the bit positi[illegible]he highest power of two found above.
- Subtract the highest power of two found above from the number.
- Repeat the process for the remaining number

Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

Another Procedure for Converting from Decimal to Binary

- Example: Converting 76d to Binary

1	.	.	.	.	.	.

  - The highest power of 2 less or equal to 76 is 64, hence the seventh (MSB) bit is 1

Assignment Project Exam Help

  - Subtracting 64 from 7

https://eduassistpro.github.io/

  - The highest power of ence the fourth bit position is 1

Add WeChat edu_assist_pro

  - We subtract 8 from 12 and get 4.
  - The highest power of 2 less or equal to 4 is 4, hence the third bit position is 1
  - Subtracting 4 from 4 yield a zero, hence all the left bits are set to 0 to yield the final answer

1	0	0	1	1	0	0

Hexadecimal Integers

- Binary values are represented in hexadecimal.

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

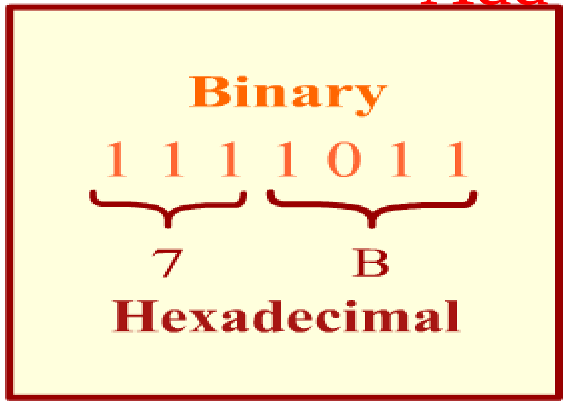
Converting Binary to Hexadecimal

- Each hexadecimal digit corresponds to 4 binary bits.
- Example: Translate the binary integer 000101101010011110010100 to hexadecimal

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro



M1023.swf

Converting Hexadecimal to Binary

- Each Hexadecimal digit can be replaced by its 4-bit binary number to form the binary equivalent.

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

M1021.swf

Converting Hexadecimal to Decimal

- Multiply each digit by its corresponding power of 16:

  Decimal = $(d3 \times 16^3) + (d2 \times 16^2) + (d1 \times 16^1) + (d0 \times 16^0)$

  d = hexadecimal digit

- Examples:
  - Hex 1234 = $(1 \times 16^3) + (2 \times 16^2) + (3 \times$ [illegible] $^0) =$ Decimal 4,660
  - Hex 3BA4 = $(3 \times 16^3) + (11 * 16^2) + (10 \times 16^1) + (4 \times 16^0) =$ Decimal 15,268

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Converting Decimal to Hexadecimal

Repeatedly divide the decimal integer by 16. Each remainder is a hex digit in the translated value:

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

least significant digit

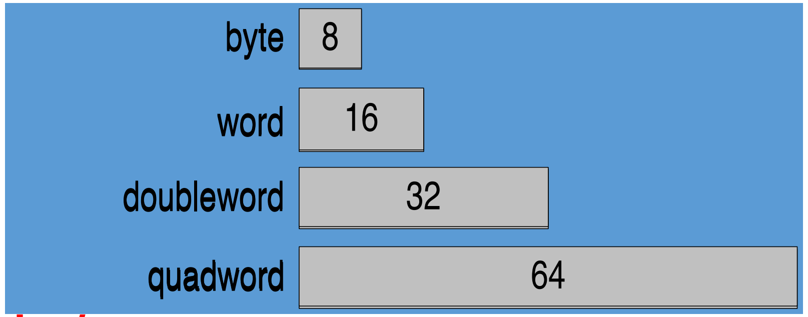
most significant digit

stop when quotient is zero

Decimal 422 = 1A6 hexadecimal

Integer Storage Sizes

Standard sizes:



Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

What is the largest unsigned integer that may be stored in 20 bits?

Binary Addition

- Start with the least significant bit (rightmost bit)
- Add each pair of bits
- Include the carry in the addition, if present

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

carry:

	0	0	0	0	0	1	0	0	(4)
+	0	0	0	0	0	1	1	1	(7)
	0	0	0	0	1	0	1	1	(11)
bit position:	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**	

Hexadecimal Addition

- Divide the sum of two digits by the number base (16). The quotient becomes the carry value, and the remainder is the sum digit.

Assignment Project Exam Help

	1	1
36		
42		
78	6D	80

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

21 / 16 = 1, remainder 5

Important skill: Programmers frequently add and subtract the addresses of variables and instructions.

Signed Integers

- Several ways to represent a signed number
  - Sign-Magnitude
  - 1's complement
  - 2's complement
- Divide the range of values into 2 eq
  - First part corresponds to the positive)
  - Second part correspond to the negative numbers (< 0)
- Focus will be on the 2's complement representation
  - Has many advantages over other representations
  - Used widely in processors to represent signed integers

Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

Two's Complement Representation

Positive numbers

Signed value = Unsigned value

Negative numbers

Signed value = Unsigned value - 2^n

Assignment Project Exam Help

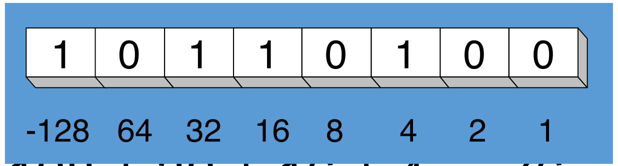
n = number of bit

https://eduassistpro.github.io/

Negative weight f

Add WeChat edu_assist_pro

Another way to obtain the signed v is to assign a negative weight to most-significant bit



= -128 + 32 + 16 + 4 = -76

8-bit Binary value	Unsigned value	Signed value
00000000	0	0
00000001	1	+1
00000010	2	+2
. . .	. . .	. . .
10	126	+126
11	127	+127
10000000	128	-128
10000001	129	-127
. . .	. . .	. . .
11111110	254	-2
11111111	255	-1

Forming the Two's Complement

starting value	`00100100 = +36`
step1: reverse the bits (1's complement)	`11011011`
step 2: add 1 to the value from step 1	`+        1`
sum = 2's compleme	`1011100 = -36`

Assignment Project Exam Help
https://eduassistpro.github.io/

Sum of an integer and its 2's c t must be zero:

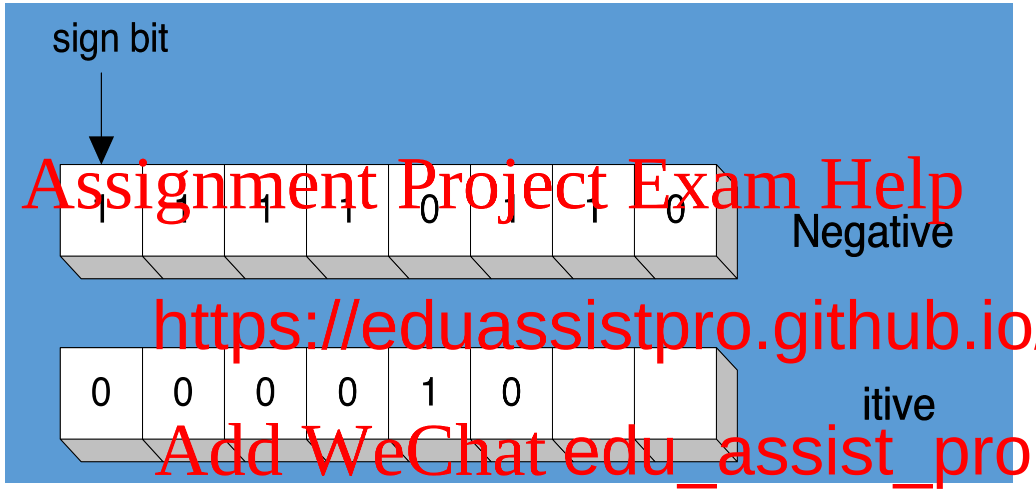
Add WeChat edu_assist_pro

00100100 + 11011100 = 000000) ⇒ Ignore Carry

The easiest way to obtain the 2's complement of a binary number is by starting at the LSB, leaving all the 0s unchanged, look for the first occurrence of a 1. Leave this 1 unchanged and complement all the bits after it.

Sign Bit

Highest bit indicates the sign. 1 = negative, 0 = positive



Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

If highest digit of a hexadecimal is > 7, the value is negative

Examples: 8A and C5 are negative bytes

A21F and 9D03 are negative words

B1C42A00 is a negative double-word

Sign Extension

Step 1: Move the number into the lower-significant bits

Step 2: Fill all the remaining higher bits with the sign bit

- This will ensure that both magnitude and sign are correct

Assignment Project Exam Help

- Examples

https://eduassistpro.github.io/

  - Sign-Extend 10110011 t

    `10110011 = -77` → `0110011 = -77`

Add WeChat edu_assist_pro

  - Sign-Extend 01100010 to 16 bits

    `01100010 = +98` → `00000000 01100010 = +98`

- Infinite 0s can be added to the left of a positive number
- Infinite 1s can be added to the left of a negative number

Two's Complement of a Hexadecimal

- To form the two's complement of a hexadecimal
  - Subtract each hexadecimal digit from 15
  - Add 1

Assignment Project Exam Help

- Examples:

https://eduassistpro.github.io/

  - 2's complement of 6A3D = 95

Add WeChat edu_assist_pro

  - 2's complement of 92F0 = 6
  - 2's complement of FFFF = 0001
- No need to convert hexadecimal to binary

Two's Complement of a Hexadecimal

- Start at the least significant digit, leaving all the 0s unchanged, look for the first occurrence of a non-zero digit.
- Subtract this digit fro
- Then subtract all rema

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

- Examples:
  - 2's complement of 6A3D = 95C3
  - 2's complement of 92F0 = 6D10
  - 2's complement of FFFF = 0001

```
      16          F F 16
- 6 A 3  D     - 9 2  F 0
-----------    -----------
  9 5 C 3        6 D 1 0
```

Binary Subtraction

- When subtracting A – B, convert B to its 2's complement
- Add A to (–B)

	0 0 0 0 1 1 0 0		0 0 0 0 1 1 0 0
–	0 0 0 0 0 0 1	+	's complement)
	0 0 0 0 1 0 1		me result)

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

- Carry is ignored, because
  - Negative number is sign-extended with 1's
  - You can imagine infinite 1's to the left of a negative number
  - Adding the carry to the extended 1's produces extended zeros

Practice: Subtract 00100101 from 01101001.

Hexadecimal Subtraction

- When a borrow is required from the digit to the left, add 16 (decimal) to the current digit's value

```
  16 + 5 = 21
        -1
    C675
  - A247
    ----
    242E
```

Assignment Project Exam Help

1 1

https://eduassistpro.github.io/

complement)

Add WeChat edu_assist_pro

result)

- Last Carry is ignored

Practice: The address of **var1** is 00400B20. The address of the next variable after var1 is 0040A06C. How many bytes are used by var1?

Ranges of Signed Integers

The unsigned range is divided into two signed ranges for positive and negative numbers

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Practice: What is the range of signed values that may be stored in 20 bits?

Carry and Overflow

- Carry is important when …
  - Adding or subtracting unsigned integers
  - Indicates that the unsigned sum is out of range
  - Either < 0 or > maximum
- Overflow is important w
  - Adding or subtracting signed integers
  - Indicates that the signed sum is out of range
- Overflow occurs when
  - Adding two positive numbers and the sum is negative
  - Adding two negative numbers and the sum is positive
  - Can happen because of the fixed number of sum bits

Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

Carry and Overflow Examples

- We can have carry without overflow and vice-versa
- Four cases are possible

				1					
	0	0	0	0	1	1	1	1	15
+	0	0	0	0	1	0			
	0	0	0	1	0	1			

Carry = 0   Overflow = 0

1	1	1	1	1					
	0	0	0	0	1	1	1	1	15
+				1	1	0	0	0	245 (-8)
				0	0	1	1	1	7

verflow = 0

	1								
	0	1	0	0	1	1	1	1	79
+	0	1	0	0	0	0	0	0	64
	1	0	0	0	1	1	1	1	143 (-113)

Carry = 0   Overflow = 1

1			1	1					
	1	1	0	1	1	0	1	0	218 (-38)
+	1	0	0	1	1	1	0	1	157 (-99)
	0	1	1	1	0	1	1	1	119

Carry = 1   Overflow = 1

Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

Character Storage

- Character sets
  - Standard ASCII: 7-bit character codes (0 – 127)
  - Extended ASCII: 8-bit character codes (0 – 255)
  - Unicode: 16-bit character codes (0 – 65,535)
  - Unicode stan al character set
    - Defines cod on languages
    - Used in Windows-XP: each cha ed as 16 bits
  - UTF-8: variable-length encodi TML
    - Encodes all Unicode characters
    - Uses 1 byte for ASCII, but multiple bytes for other characters
- Null-terminated String
  - Array of characters followed by a NULL character

Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

ASCII Codes

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Examples:

- ASCII code for space character = 20 (hex) = 32 (decimal)
- ASCII code for ‘A' = 41 (hex) = 65 (decimal)
- ASCII code for 'a' = 61 (hex) = 97 (decimal)

Control Characters

- The first 32 characters of ASCII table are used for control
- Control character codes = 00 to 1F (hex)

Assignment Project Exam Help

- Examples of Control Char
  - Character 0 is the NULL ch a string

https://eduassistpro.github.io/

  - Character 9 is the Horizontal Tab (HT) charac

Add WeChat edu_assist_pro

  - Character 0A (hex) = 10 (decimal) is the Line
  - Character 0D (hex) = 13 (decimal) is the Carriage Return (CR)
  - The LF and CR characters are used together
    - They advance the cursor to the beginning of next line
- One control character appears at end of ASCII table
  - Character 7F (hex) is the Delete (DEL) character

Parity Bit

- Data errors can occur during data transmission or storage/retrieval.
- The 8th bit in the ASCII code is used for error checking.
- This bit is usually referred to as the parity bit.

Assignment Project Exam Help

- There are two ways fo

https://eduassistpro.github.io/

  - Even Parity: Where the 8th bit is set su total number of 1s in the 8-bit code word is even.

Add WeChat edu_assist_pro

  - Odd Parity: The 8th bit is set such that the total number of 1s in the 8-bit code word is odd.

P							
1	1	0	0	0	0	0	1