**Assignment No.11**

|  |
| --- |
| **Title of Assignment:**  Write 80387 ALP to find sine wave, Cosine wave and Sinc function |
| **Relevant Theory:**  **Explanation:**  **a) Features of 80387:**  • High performance 80-Bit Internal Architecture  • Implements ANSI/IEEE standard 754-1985 for Binary floating-point arithmetic  • Expands Intel386DX CPU data types to include 32-, 64-, 80-bit floating point, 32-, 64-bit  integers and 18-bit BCD operands  • Extends Intel386DX CPU instruction set to include Trigonometric, Logarithmic,  Exponential and Arithmetic instructions for all data types  • Upward object code compatible  • Full-range transcendental operations for SINE, COSINE, TANGENT, ARCTANGENT  and LOGARITHM  • Built-in Exception handling  • Operates independently in all modes of 80386  • Eight 80-bit Numeric registers  • Available in 68-pin PGA package  • One version supports 16MHz-33MHz  **80387 Functional block diagram:**    Figure . Intel387TM DX Math CoProcessor Block Diagram  **Register Set**   * Data registers: Eight 80-bit registers, * Tag Word: the tag word marks the content of each numeric data register, two bits for each data register * Status word: the 16-bit status word reflects the overall state of the MCP * Instruction and Data pointers: two pointer registers allows identification of the   failing numeric instruction which supply the address of failing numeric instruction  and the address of its numeric memory operand.   * Control Word: several processing options are selected by loading a control word   from memory into the control register  c) Instruction of co-processor used in the assignment:  FINIT: Initialise Co-processor  FLDZ: Load zero on stack top  FILD: Load Integer on stack  FIDIV: Divide stack top by an integer value  FIMUL: Multiply stack top by an integer value  FST: Store stack top  FADD: Add in stack top  FBSTP: Store integer part of stack top in 10 byte packed BCD format  FMUL: Multiply stack top  FSQRT: Square Root of Stack Top  FSTSW: Stores the coprocessor status word  FTS: compares ST0 and 0  FSIN: calculate sine value  FCOS: calculate COS value  DOS interrupts used inthis program   1. int 21h : call the interrupt handler 0x21 which is the DOS Function dispatcher. 2. **AH = 01h - READ CHARACTER FROM STANDARD INPUT, WITH ECHO** 3. **AH = 09h - WRITE STRING TO STANDARD OUTPUT**   Entry: DS:DX -> '$'-terminated string  Return: AL = 24h   1. **AH = 4Ch - "EXIT" - TERMINATE WITH RETURN CODE**   Entry: AL = return code  Return: never returns  **Int 10H: set video BIOS services.** INT 10,0 - Set Video Mode AH = 00  AL = 00 40x25 B/W text (CGA,EGA,MCGA,VGA)  = 01 40x25 16 color text (CGA,EGA,MCGA,VGA)  = 02 80x25 16 shades of gray text (CGA,EGA,MCGA,VGA)  = 03 80x25 16 color text (CGA,EGA,MCGA,VGA)  = 04 320x200 4 color graphics (CGA,EGA,MCGA,VGA)  = 05 320x200 4 color graphics (CGA,EGA,MCGA,VGA)  = 06 640x200 B/W graphics (CGA,EGA,MCGA,VGA)  = 07 80x25 Monochrome text (MDA,HERC,EGA,VGA)  = 08 160x200 16 color graphics (PCjr)  = 09 320x200 16 color graphics (PCjr)  = 0A 640x200 4 color graphics (PCjr)  = 0B Reserved (EGA BIOS function 11)  = 0C Reserved (EGA BIOS function 11)  = 0D 320x200 16 color graphics (EGA,VGA)  = 0E 640x200 16 color graphics (EGA,VGA)  = 0F 640x350 Monochrome graphics (EGA,VGA)  = 10 640x350 16 color graphics (EGA or VGA with 128K)  640x350 4 color graphics (64K EGA)  = 11 640x480 B/W graphics (MCGA,VGA)  = 12 640x480 16 color graphics (VGA)  = 13 320x200 256 color graphics (MCGA,VGA)  = 8x EGA, MCGA or VGA ignore bit 7, see below  = 9x EGA, MCGA or VGA ignore bit 7, see below  - if AL bit 7=1, prevents EGA,MCGA & VGA from clearing display  - function updates byte at 40:49; bit 7 of byte 40:87  (EGA/VGA Display Data Area) is set to the value of AL bit 7 INT 10,C - Write Graphics Pixel at Coordinate AH = 0C  AL = color value (XOR'ED with current pixel if bit 7=1)  BH = page number, see [VIDEO PAGES](http://stanislavs.org/helppc/video_pages.html)  CX = column number (zero based)  DX = row number (zero based)  returns nothing  - if bit 7 is 1, color specified is XOR'ed with current pixel  - page number in BH ignored for 320x200 4 color graphics mode  - this function is known to destroy AX and possibly SI and DI on  on some PS/2 VGA systems INT 16 - Keyboard BIOS ServicesINT 16,0 - Wait for Keypress and Read Character AH = 00  on return:  AH = keyboard scan code  AL = ASCII character or zero if special function key  - halts program until key with a scancode is pressed    **d) Concept of sine wave, cosine wave, sinc wave**  **Sine wave: To draw sine wave we need to calculate value of function sin(X).**  **x is input for to calculate sine wave.**  **Convert it into radian**  **Then fine sine value.**  **That is sine((pi/180)\*x)**  **Cosine wave: To draw sine wave we need to calculate value of function cos(X).**  **x is input for to calculate cos wave.**  **Convert it into radian**  **Then fine cos value.**  **That is cos((pi/180)\*x**  **Sinc function: sinc(x)=sinx/x.**  **Here first calculate sine function then divide it by x.**  **Design Analysis/ Implementation Logic:**  **Algorithm**  Write here algorithm in your own words. |
| **Testing:**  **Test Conditions:**  **Input:x value.**  **Output:**  Sine wave ,Cos wave, Sinc function |
| **FAQs:**   1. Explain int 21h 2. Explain int 10 h 3. Explain int 16 h. |
| **Conclusion:** Successfully implemented 80387 ALP to find the mean, variance, and standard deviation.  . |