**Sample Programs**

* Data movement operations

|  |  |
| --- | --- |
| **SIC** | **SIC/XE** |
| LDA M  STA N  LDCH P  STCH Q   1. WORD 5 2. RESW 1 3. BYTE C’Z’ 4. RESB 1 | LDA #5  STA N  LDA #90  STCH Q  N RESW 1  Q RESB 1 |

* Write a program to compute the following
* BETA = (ALPHA+INCR -1)
* DELTA = (GAMMA + INCR - 1)

|  |  |
| --- | --- |
| **SIC** | **SIC/XE** |
| LDA ALPHA  ADD INCR  SUB ONE  STA BETA  LDA GAMMA  ADD INCR  SUB ONE  STA DELTA  ALPHA WORD 5  GAMMA WORD 8  INCR WORD 6  ONE WORD 1  BETA RESW 1  DELTA RESW 1 | LDS INCR  LDA ALPHA  ADDR S,A  SUB #1  STA BETA  LDA GAMMA  ADDR S,A  SUB #1  STA DELTA  ALPHA WORD 5  GAMMA WORD 8  INRC WORD 6  BETA RESW 1  DELTA RESW 1 |

* Write a program to copies 11 bit character string from one string to another

|  |  |
| --- | --- |
| **SIC** | **SIC/XE** |
| LDX ZERO  J1 LDCH STR1,X STCH STR2,X  TIX ELEVEN  JLT J1  ZERO WORD 0  ELEVEN WORD 11  STR1 BYTE C’TEST STRING’ STR2 RESB 11 | LDT #11  LDX #0  J1 LDCH STR1,X STCH STR2,X  TIXR T  JLT J1  STR1 BYTE C’TEST STRING’ STR2 RESB 11 |

* Write a program to do the following
  + Gamma[] = Alpha[] + Beta[]

|  |  |
| --- | --- |
| **SIC** | **SIC/XE** |
| LDA ZERO  STA INDEX  LOOP LDX INDEX  LDA ALPHA,X  ADD BETA,X  STA GAMMA,X  LDA INDEX  ADD THREE  STA INDEX  COMP K300  JLT LOOP  ZERO WORD 0  THREE WORD 3  K300 WORD 300  INDEX RESW 1  ALPHA RESW 100  BETA RESW 100  GAMMA RESW 100 | LDS #3  LDT #300  LDX #0  LOOP LDA ALPHA,X  ADD BETA,X  STA GAMMA,X  ADDR S,X  COMPR X,T  JLT LOOP  ALPHA RESW 100  BETA RESW 100  GAMMA RESW 100 |

Write SIC and SIC/XE program to calculate ALPHA = BETA x GAMMA.

|  |  |
| --- | --- |
| **SIC** | **SIC/XE** |
| LDA BETA  MUL GAMMA  STA ALPHA  ALPHA RESW 1  BETA  WORD  10  GAMMA WORD 20 | LDA BETA  LDS GAMMA  MULR S,A  STA ALPHA  ALPHA RESW 1  BETA  WORD  10  GAMMA WORD 20 |

Write SIC and SIC/XE program to set ALPHA = 4\*BETA–9.

|  |  |
| --- | --- |
| **SIC** | **SIC/XE** |
| LDA BETA  MUL FOUR  SUB NINE  STA ALPHA  ALPHA RESW 1  BETA WORD 10  FOUR WORD 4  NINE WORD 9 | LDA BETA  LDS #4  MULR S,A  SUB #9  STA ALPHA  ALPHA RESW 1  BETA WORD 10 |

Write SIC and SIC/XE program to swap the values of ALPHA and BETA.

|  |  |
| --- | --- |
| **SIC** | **SIC/XE** |
| LDA ALPHA  STA GAMMA  LDA BETA  STA ALPHA  LDA GAMMA  STA BETA  ALPHA WORD 10  BETA  WORD 20  GAMMA RESW 1 | LDA ALPHA  LDS BETA  STA BETA  STS ALPHA  ALPHA WORD 10  BETA  WORD 20 |

Write SIC and SIC/XE program to set ALPHA = integer portion of (BETA ÷ GAMMA)

|  |  |
| --- | --- |
| **SIC** | **SIC/XE** |
| LDA BETA  DIV GAMMA  STA ALPHA  ALPHA RESW 1  BETA  WORD  20  GAMMA WORD 15 | LDA BETA  LDS GAMMA  DIVR S,A  STA ALPHA  ALPHA RESW 1  BETA  RESW  20  GAMMA RESW 15 |

Write SIC and SIC/XE program to set ALPHA= integer portion of (BETA /GAMMA) and DELTA=remainder of(BETA /GAMMA).

|  |  |
| --- | --- |
| **SIC** | **SIC/XE** |
| LDA BETA  DIV GAMMA  STA ALPHA  LDA GAMMA  MUL ALPHA  STA TEMP  LDA BETA  SUB TEMP  STA DELTA  BETA  WORD  20  GAMMA WORD  7  ALPHA RESW 1  DELTA RESW 1  TEMP RESW 1 | LDA BETA  LDS GAMMA  DIVR  S, A  STA ALPHA  MULR S, A  LDS BETA  SUBR A, S  STS DELTA  BETA  WORD  20  GAMMA WORD  7  ALPHA RESW 1  DELTA RESW 1 |

Write a sequence of instructions for SIC/XE to set ALPHA= nearest integer of (BETA/GAMMA). Use reg F for calculation.

|  |
| --- |
| **SIC/XE** |
| LDF BETA  DIVF GAMMA  FIX  STA ALPHA  ALPHA RESW 1  BETA  WORD  20  GAMMA WORD 7 |

Write SIC and SIC/XE program to clear a 20-byte string to all blanks

|  |  |
| --- | --- |
| **SIC** | **SIC/XE** |
| LDX ZERO  LDCH BLANK  LOOP STCH STR1,X  TIX TWENTY  JLT LOOP  STR1  RESB  20  BLANK BYTE C ‘’  ZERO WORD 0  TWENTY WORD 20 | LDX #0  LDS #20  LDCH #0  LOOP STCH STR1,X  TIXR S  JLT LOOP  STR1  RESB  20 |

Suppose that ALPHA is an array of 100 words. Write SIC and SIC/XE program to set all 100 elements of the array to 0.

|  |  |
| --- | --- |
| **SIC** | **SIC/XE** |
| LDA ZERO  STA INDEX  LOOP  LDX  INDEX  LDA ZERO  STA  ALPHA, X  LDA INDEX  ADD  THREE  STA INDEX  COMP K300  JLT LOOP  INDEX RESW 1  ALPHA RESW 100  ZERO WORD 0  K300  WORD  300  THREE WORD 3 | LDS #3  LDT #300  LDX #0  LOOP LDA #0  STA  ALPHA, X  ADDR  S, X  COMPR X, T  JLT LOOP  ALPHA RESW 100 |

**TUTORIAL 4**

Suppose that ALPHA and BETA are the two arrays of 100 words. Another array of GAMMA elements are obtained by multiplying the corresponding ALPHA element by 4 and adding the corresponding BETA elements. Write SIC and SIC\XE program.

|  |  |
| --- | --- |
| **SIC** | **SIC/XE** |
| LOOP LDX INDEX  LDA ALPHA, X  MUL FOUR  ADD  BETA, X  STA  GAMMA, X  LDA INDEX  ADD THREE  STA INDEX  COMP K300  JLT LOOP  ALPHA RESW 100  BETA  RESW  100  GAMMA RESW 100  FOUR WORD 4  THREE WORD 3  K300 WORD 300  INDEX WORD 0 | LDS #3  LDT #300  LDX #0  LOOP LDA ALPHA, X  MUL #4  ADD  BETA, X  STA  GAMMA, X  ADDR  S, X  COMPR X, T  JLT LOOP  ALPHA RESW 100  BETA  RESW  100  GAMMA RESW 100 |

Suppose that ALPHA is an array of 100 words. Write SIC and SIC/XE program to find the maximum element in the array and store results in MAX

|  |  |
| --- | --- |
| **SIC** | **SIC/XE** |
| LDA ZERO  STA INDEX  LOOP1 LDX INDEX  LDA ALPHA, X  COMP MAX  JLT LOOP2  STA MAX  LOOP2 LDA INDEX  ADD THREE  STA INDEX  COMP K300  JLT LOOP1  ZERO WORD 0  ALPHA RESW 100  K300 WORD 300  INDEX WORD 0  THREE WORD 3  MAX WORD -32768 | LDS #3  LDT #300  LDX #0  LOOP1 LDA ALPHA, X  COMP MAX  JLT LOOP2  STA MAX  LOOP2 ADDR S, X  COMPR X, T  JLT LOOP1  ALPHA RESW 100  MAX WORD -32768 |

Suppose that RECORD contains a 100-byte record. Write SIC and SIC/XE program to write this record on to device 05.

|  |  |
| --- | --- |
| **SIC** | **SIC/XE** |
| JSUB WRREC  WRREC LDX ZERO  LOOP TD OUTPUT  JEQ LOOP  LDCH  RECORD, X  WD OUTPUT  TIX LENGTH  JLT LOOP  RSUB  ZERO WORD 0  LENGTH WORD 100  OUTPUT BYTE X,’05’  RECORD RESB 100 | JSUB WRREC  WRREC LDX #0  LDT #100  LOOP TD OUTPUT  JEQ LOOP  LDCH  RECORD, X  WD OUTPUT  TIXR T  JLT LOOP  RSUB  OUTPUT BYTE X ,’05’  RECORD RESB 100 |

Write a subroutine for SIC and SIC/XE that will read a record into a buffer. The record may be any length from 1 to 100 bytes. The end of record is marked with a “null” character (ASCII code 00). The subroutine should place the length of the record read into a variable named LENGTH

|  |  |
| --- | --- |
| **SIC** | **SIC/XE** |
| JSUB RDREC  .  .  RDREC LDX ZERO  LOOP TD INDEV  JEQ LOOP  RD INDEV  COMP NULL  JEQ EXIT  STCH  BUFFER, X  TIX K100  JLT RLOOP  EXIT  STX  LENGTH  RSUB  ZERO WORD 0  NULL  WORD  0  K100  WORD  100  INDEV BYTE X ,’F1’  LENGTH RESW 1  BUFFER RESB 100 | JSUB RDREC  .  .  RDREC  LDX   #0  LDT #100  LDS #0  LOOP TD INDEV  JEQ LOOP  RD INDEV  COMPR A, S  JEQ EXIT  STCH  BUFFER, X  TIXR T  JLT RLOOP  EXIT STX LENGTH  RSUB  INDEV BYTE X,’F1’  LENGTH RESW 1  BUFFER RESB 100 |

* Sample input and output operations

|  |
| --- |
| **SIC, SIC/XE** |
| LOOP1 TD INDEV  JEQ LOOP1  RD INDEV STCH DATA  LOOP2 TD OUTDEV  JEQ LOOP2 LDCH DATA WD OUTDEV  INDEV BYTE X’F1’ OUTDEV BYTE X’05’  DATA RESB 1 |

* Write a SIC program to Read 100 byte record into buffer using subroutine

|  |  |
| --- | --- |
| **SIC** | **SIC/XE** |
| JSUB READ  READ LDX ZERO LOOP1 TD INDEV  JEQ LOOP1  RD INDEV STCH RECORD,X TIX K100  JLT LOOP1 RSUB  INDEV BYTE X’F1’ RECORD RESB 100  ZERO WORD 0  K300 WORD 300 | JSUB READ  READ LDX #0  LDT #100 LOOP1 TD INDEV  JEQ LOOP1  RD INDEV STCH RECORD,X TIXR T  JLT LOOP1 RSUB  INDEV BYTE X’F1’ RECORD RESB 100 |