



Quiz 4 / Fall 2012

Even

Q1. Write a program in Assembly, using the annexed instruction list, which counts how many 1s are present in a 16-bit word in memory. The word's value is (FEDC)₁₆ and is found at address A00. The result is to be stored in the memory location with address B00. Your program should be stored in the memory of the basic computer starting with address 400.

NOTE: All numbers in the above text are in hexadecimal.

Solution 1:

Circulate left WRD through E to test the word msb; increment CTR if E=1 then delete E.

Redo the loop and stop when no 1's are in WRD = 0; CTR will show how many 1's were in the word.

```

    ORG 400
400    CLE          /Clear E
401    CLA          /Clear AC
402    STA CTR      /Initialize counter to 0
403    LDA WRD      /Load "word" HEX FEDC into AC
404    SZA          /Skip if current "word" in AC = 0
405    BUN ROT      /Branch to ROT (407) to start detecting and counting 1's
406    BUN STP      /"word" is zero, stop with CTR = 0
407    ROT, CIL     /Bring bit to be tested to E
408    SZE          /Skip if bit in E = 0
409    BUN AGN      /if bit = 1, count it
40A    BUN ROT      /if bit = 0, repeat the test for next bit
40B    AGN, CLE     /Clear E
40C    ISZ CTR      /Increment counter
40D    SZA          /Check if remaining bits = 0
40E    BUN ROT      /No, then rotate again
40F    STP, HLT     /Yes, then halt

    ORG A00
A00    WRD, HEX FEDC /Original word

    ORG B00
B00    CTR, HEX 0    /Final result
        END
  
```

Solution 2:

An extra condition to leave the loop is added to the previous solution: CTB counts from -16 to 0, to keep track of the tested bits.

/ Note There are 2 counters employed here:

/ CTR = counts how many bits are 1

/ CTB = counts the bits of the word, starting from -16, as in DO {...} WHILE CTR <0

ORG 400

```

400      CLE          /Clear E
401      CLA          /Clear AC
402      STA CTR      /Initialize CTR (counter of 1-bits) to 0
403      LDA N16     /Initialize CTB (counter of the word's bits) to -16
404      STA CTB
405      LDA WRD     /Load "word" HEX FEDC into AC
406      SZA          /Skip if "word" in AC = 0, i.e., the initial word does not have a single 1!
407      BUN ROT     /Branch to ROT (409) to start detecting and counting 1's
408      BUN STP     /"word" is zero, so there is no point to search for 1's; stop with CTR = 0
409  ROT,  CIL       /Bring msb of the word (in AC) to E, to be tested; the shifted word should
                    remain in AC
40A      SZE          /Skip if bit in E = 0
40B      BUN AGN     /if bit = 1, count it
40C      ISZ CTB     / Increment CTB (counter of the word's bits) until reaches 0
40D      BUN ROT     / repeat the loop, i.e., test the next bit of the "word"
40E      BUN STP     /stop since finished testing all 16 bits of the word
40F  AGN,  CLE       /Clear E
410      ISZ CTR     /Increment counter
411      SZA          /Check if remaining bits = 0
412      BUN ROT     /No, then rotate again
413  STP,  HLT       /Yes, then halt

                    /ORG A00
A00  WRD,  HEX FEDC /Original word

                    /ORG B00
B00  CTR,  HEX 0     /Final result
      CTB,
      N16,  DEC-16   Number of the word bits, to initialize CTR (there is no immediate addressing
                    mode in the Basic Computer!)

      END

```

Any other correct equivalent solution is accepted.

Symbol	Hex code	Description
AND	0 or 8	AND M to AC
ADD	1 or 9	Add M to AC, carry to E
LDA	2 or A	Load AC from M
STA	3 or B	Store AC in M
BUN	4 or C	Branch unconditionally to m
BSA	5 or D	Save return address in m and branch to m + 1
ISZ	6 or E	Increment M and skip if zero
CLA	7800	Clear AC
CLE	7400	Clear E
CMA	7200	Complement AC
CME	7100	Complement E
CIR	7080	Circulate right E and AC
CIL	7040	Circulate left E and AC
INC	7020	Increment AC
SPA	7010	Skip if AC is positive
SNA	7008	Skip if AC is negative
SZA	7004	Skip if AC is zero
SZE	7002	Skip if E is zero
HLT	7001	Halt computer
INP	F800	Input information and clear flag
OUT	F400	Output information & clear flag
SKI	F200	Skip if input flag is on
SKO	F200	Skip if output flag is on
ION	F080	Turn interrupt on
IOF	F040	Turn interrupt off