Homework 6

Computer Systems

Jacob Howard

Jah0147

**Start of Code**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Freescale HC12-Assembler

(c) Copyright Freescale 1987-2015

Abs. Rel. Loc Obj. code Source line

---- ---- ------ --------- -----------

1 1 ;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

2 2 ;\* This stationery serves as the framework for a \*

3 3 ;\* user application (single file, absolute assembly application) \*

4 4 ;\* For a more comprehensive program that \*

5 5 ;\* demonstrates the more advanced functionality of this \*

6 6 ;\* processor, please see the demonstration applications \*

7 7 ;\* located in the examples subdirectory of the \*

8 8 ;\* Freescale CodeWarrior for the HC12 Program directory \*

9 9 ;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

10 10

11 11 ; export symbols

12 12 XDEF Entry, \_Startup ; export 'Entry' symbol

13 13 ABSENTRY Entry ; for absolute assembly: mark this as application entry point

14 14

15 15

16 16

17 17 ; Include derivative-specific definitions

18 18

19 19 0000 0800 RAMStart EQU $0800

20 20 0000 4000 ROMStart EQU $4000 ; absolute address to place my code/constant data

21 21

22 22 ; variable/data section

23 23

24 24 ORG RAMStart

25 25

26 26 a000800 TABLE DS.B 1

27 27 a000801 01 DIRECTION DC.B 1

28 28 a000802 0100 BOUNCE DC.B 1,0

29 29 a000804 00 COUNTER DC.B 0 ;Will be used to check if the loop should continue

30 30

31 31

32 32

33 33 ORG ROMStart

34 34

35 35 Entry:

36 36 \_Startup:

37 37 a004000 CE08 02 ldx #BOUNCE

38 38

39 39 a004003 C608 ldab #$08

40 40 a004005 7B08 00 stab TABLE

41 41

42 42 loop:

43 43 a004008 8114 cmpa #20 ;compairs a with 20 to check if the loop should keep going

44 44 a00400A 270B beq done

45 45

46 46 a00400C 070A bsr update\_table ;Branches to subRoutine

47 47

48 48 a00400E B608 04 ldaa COUNTER ;Loads value in COUNTER

49 49 a004011 42 inca ;Increments a

50 50 a004012 7A08 04 staa COUNTER ;Stores new value into COUNTER

51 51 a004015 20F1 bra loop ;Branches back to loop

52 52

53 53 done:

54 54 a004017 3D rts

55 55

56 56

57 57

58 58 update\_table: ;update Table subRoutine

59 59

60 60 a004018 F708 00 tst TABLE ;Tests table to see if a ball is there

61 61 a00401B 2702 beq setBall ;If no ball is there, it will go to setBall

62 62 a00401D 260C bne ballLocation ;if a ball is there, it will check which direction to change the ball

63 63

64 64 setBall: ;This will set the ball at position 1 if there is no ball

Freescale HC12-Assembler

(c) Copyright Freescale 1987-2015

Abs. Rel. Loc Obj. code Source line

---- ---- ------ --------- -----------

65 65 a00401F 8601 ldaa #1

66 66 a004021 C600 ldab #0

67 67 a004023 7A08 00 staa TABLE

68 68 a004026 7B08 01 stab DIRECTION

69 69 a004029 2026 bra shiftDirection

70 70

71 71 ballLocation: ;Checks ball Location

72 72

73 73 a00402B 8601 ldaa #1

74 74 a00402D C680 ldab #128

75 75

76 76 a00402F B108 00 cmpa TABLE

77 77 a004032 2707 beq checkBounce

78 78

79 79 a004034 F108 00 cmpb TABLE

80 80 a004037 2711 beq changeDirection2

81 81

82 82 a004039 2616 bne shiftDirection

83 83

84 84 checkBounce: ;Checks Bounce

85 85 a00403B 8601 ldaa #1

86 86 a00403D A130 cmpa 1,x+

87 87 a00403F 2702 beq changeDirection1

88 88

89 89 a004041 260E bne shiftDirection

90 90

91 91

92 92 ;changeDirection1 and 2 Changes Direction of ball based on if the ball is at the start or end of the table

93 93 changeDirection1:

94 94 a004043 8600 ldaa #0

95 95 a004045 7A08 01 staa DIRECTION

96 96 a004048 2007 bra shiftDirection

97 97

98 98 changeDirection2:

99 99 a00404A 8601 ldaa #1

100 100 a00404C 7A08 01 staa DIRECTION

101 101 a00404F 2000 bra shiftDirection

102 102

103 103

104 104 shiftDirection:

105 105 a004051 C601 ldab #1

106 106 a004053 F108 01 cmpb DIRECTION

107 107 a004056 2602 bne shiftLeft

108 108 a004058 2704 beq shiftRight

109 109

110 110 shiftLeft: ;Shifts ball to Left

111 111 a00405A 7808 00 lsl TABLE

112 112 a00405D 3D rts

113 113

114 114

115 115 shiftRight: ;Shifts ball to Right

116 116 a00405E 7408 00 lsr TABLE

117 117 a004061 3D rts

118 118

119 119

120 120

121 121

122 122

123 123 ; result in D

124 124

125 125 ;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

126 126 ;\* Interrupt Vectors \*

127 127 ;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

128 128 ORG $FFFE

Freescale HC12-Assembler

(c) Copyright Freescale 1987-2015

Abs. Rel. Loc Obj. code Source line

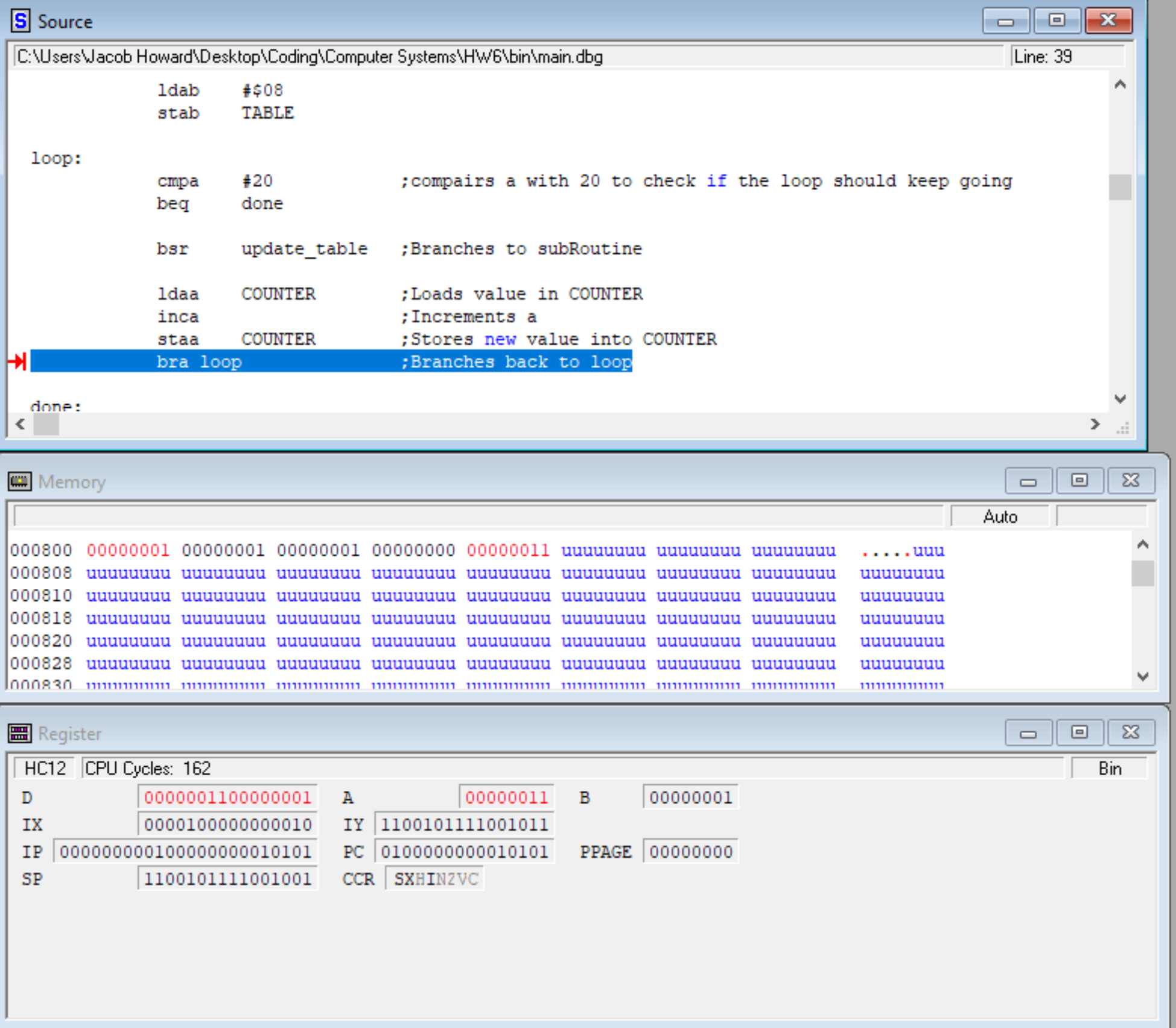
---- ---- ------ --------- -----------

129 129 a00FFFE 4000 DC.W Entry ; Reset Vector

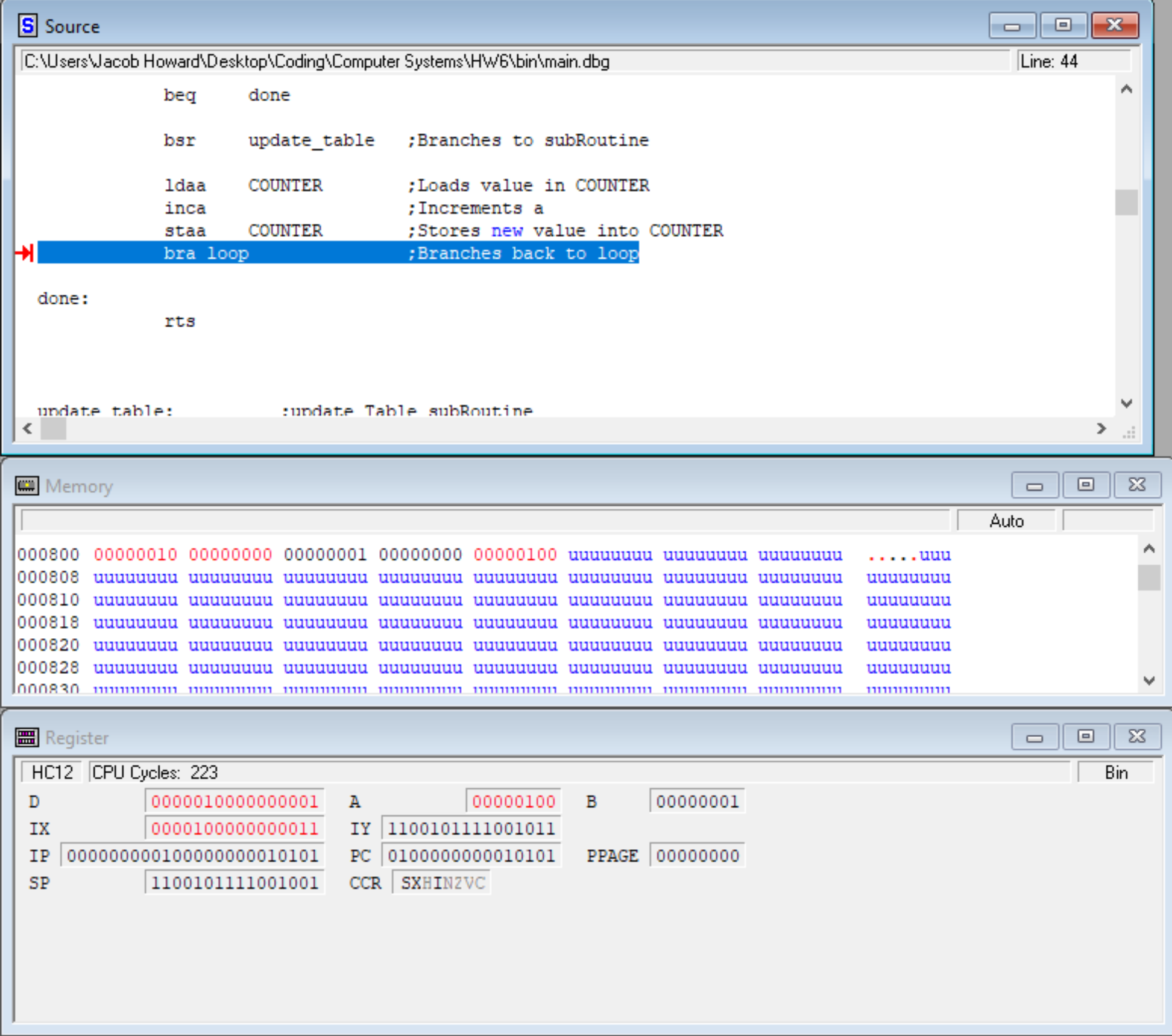
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**End of Code**

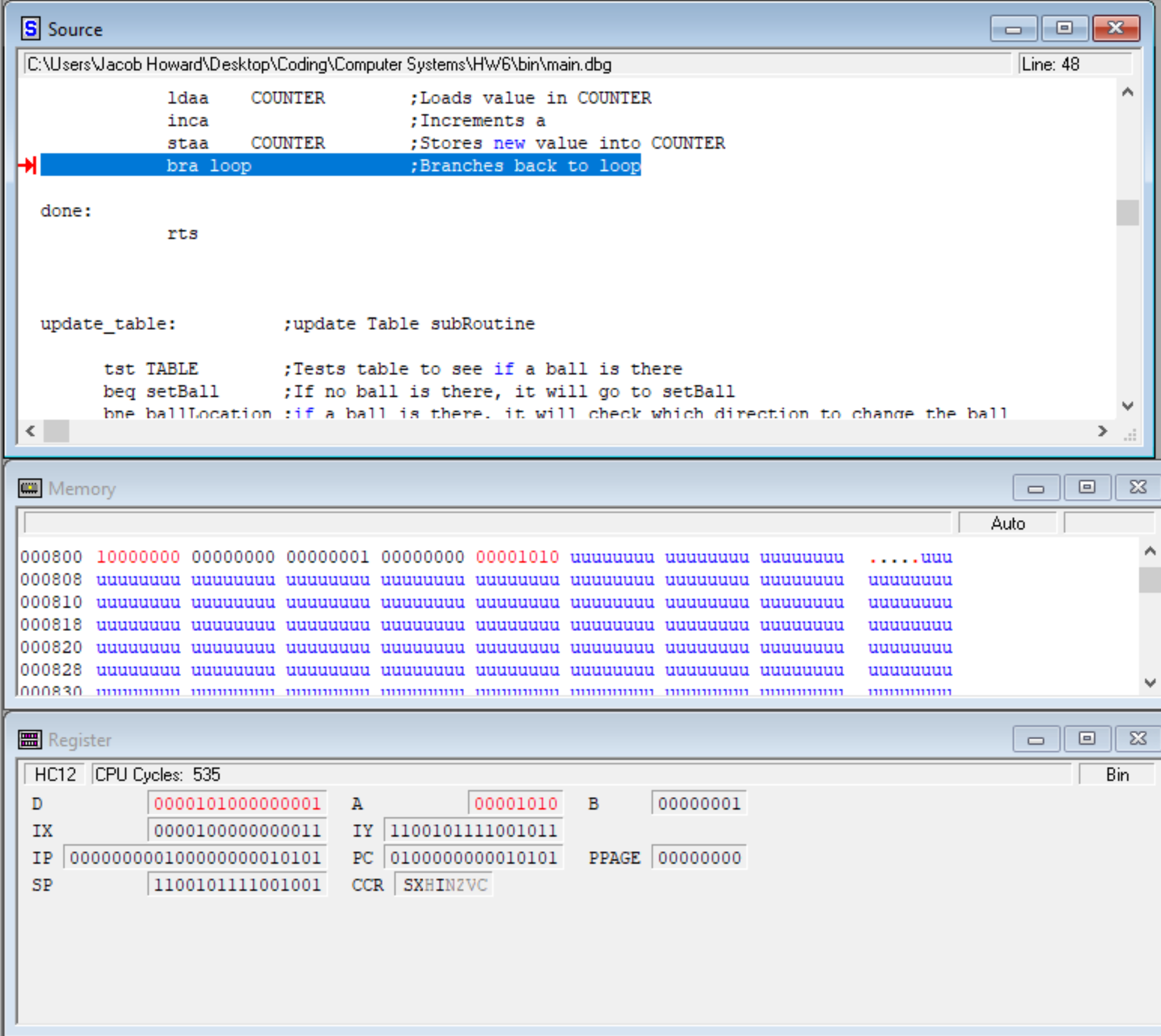
**3rd Step**



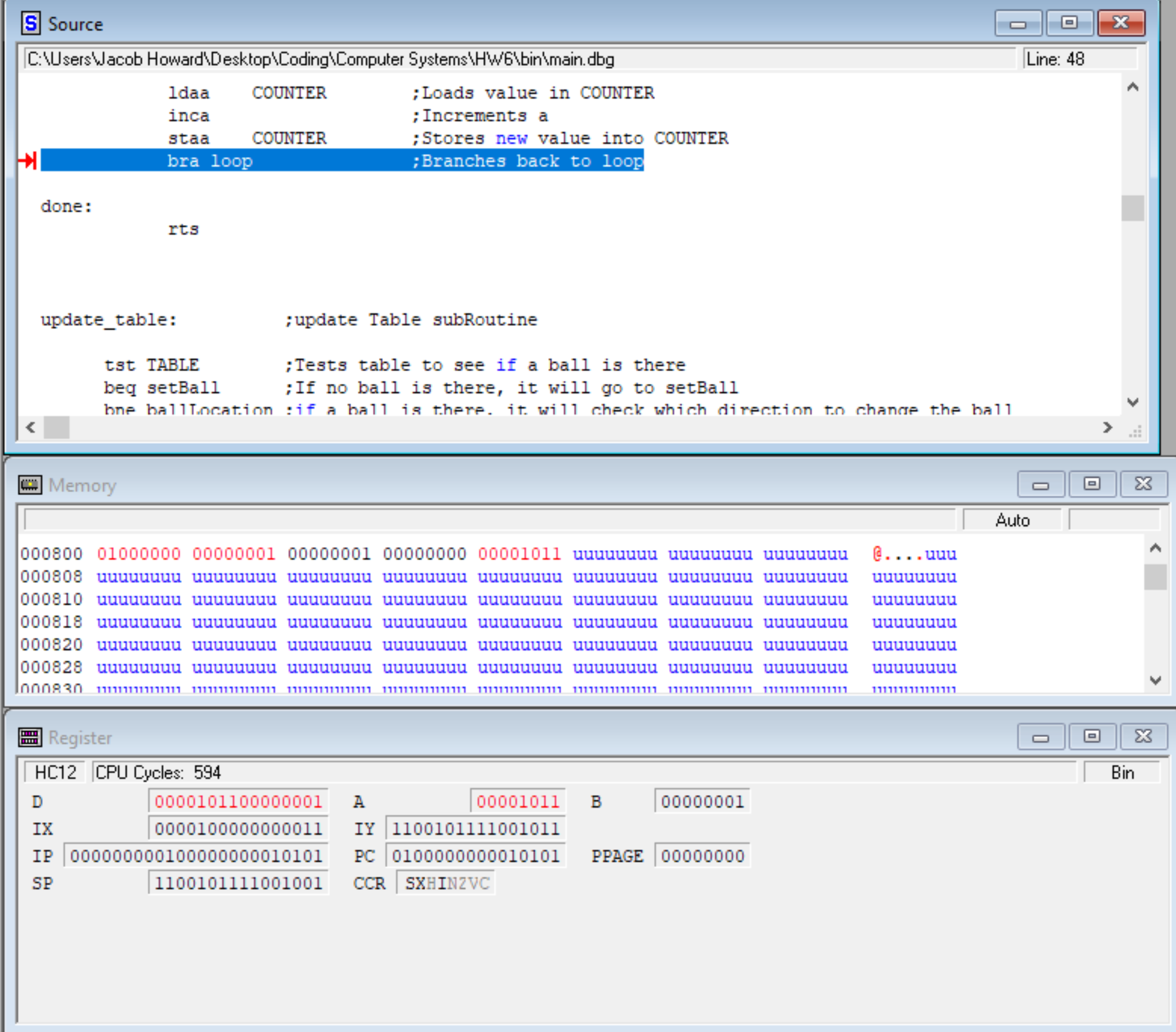
**4th Step**



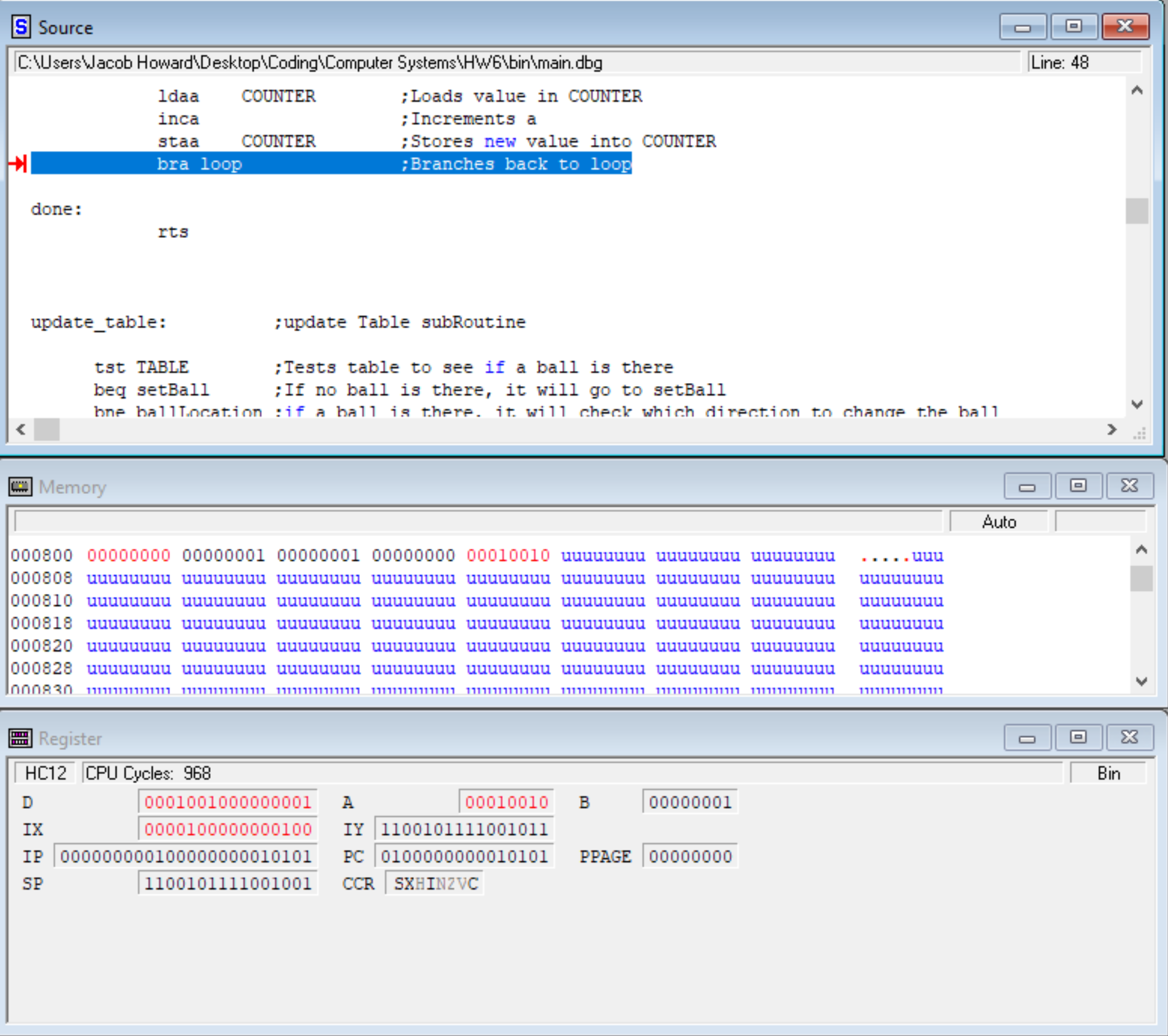
**10th step**



**11th step**



**18th Step**



**20th Step**

**(On the 9th step, the Ball is placed back at the 00000001 position and the code continues to shift the ball to the left after to the 00000010 position. That is why at the 20th step, when the ball is shifted it is at the 00000100 position)**

