Candace Caimol

CPE301 – SPRING 2016

Design Assignment X

**DO NOT REMOVE THIS PAGE DURING SUBMISSION:**

The student understands that all required components should be submitted in complete for grading of this assignment.

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **SUBMISSION ITEM** | **COMPLETED (Y/N)** | **MARKS**  **(/MAX)** |
| 0. | COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS |  |  |
| 1. | INITIAL CODE OF TASK 1/A |  |  |
| 2. | INCREMENTAL / DIFFERENTIAL CODE OF TASK 2/B |  |  |
| 3. | INCREMENTAL / DIFFERENTIAL CODE OF TASK 3/C |  |  |
| 4. | INCREMENTAL / DIFFERENTIAL CODE OF TASK 4/D |  |  |
| 5. | INCREMENTAL / DIFFERENTIAL CODE OF TASK 5/E |  |  |
| 6. | SCHEMATICS |  |  |
| 7. | SCREENSHOTS OF EACH TASK OUTPUT |  |  |
| 8. | SCREENSHOT OF EACH DEMO |  |  |
| 9. | VIDEO LINKS OF EACH DEMO |  |  |
| 10. | GOOGLECODE LINK OF THE DA |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 1. | CODE OF TASK A |  |  |

;

; DA0.asm

;

; Created: 2/14/2016 12:25:46 PM

; Author : user

;

; Replace with your application code

.org 0

SBI DDRB, 2 ;set PB2 as output

LDI R18, 0 ;output register set to PB2 = 0 (off)

LDI R16, 55 ;load immediate to R16 (NUM1)

LDI R17, 55 ;load immediate to R17 (NUM2)

ADD R17, R16 ;sum = NUM2 + NUM1

LDI R16, 55 ;load immediate to R16 (NUM3)

ADD R17, R16 ;sum = sum + NUM3

LDI R16, 55 ;load immediate to R16 (NUM4)

ADD R17, R16 ;sum = sum + NUM4

LDI R16, 55 ;load immediate to R16 (NUM5)

ADD R17, R16 ;sum = sum + NUM5

BRCC NoOverflow ;if no carry, branch to where light won't turn on

LDI R18, 4 ;else set to light up bit 2

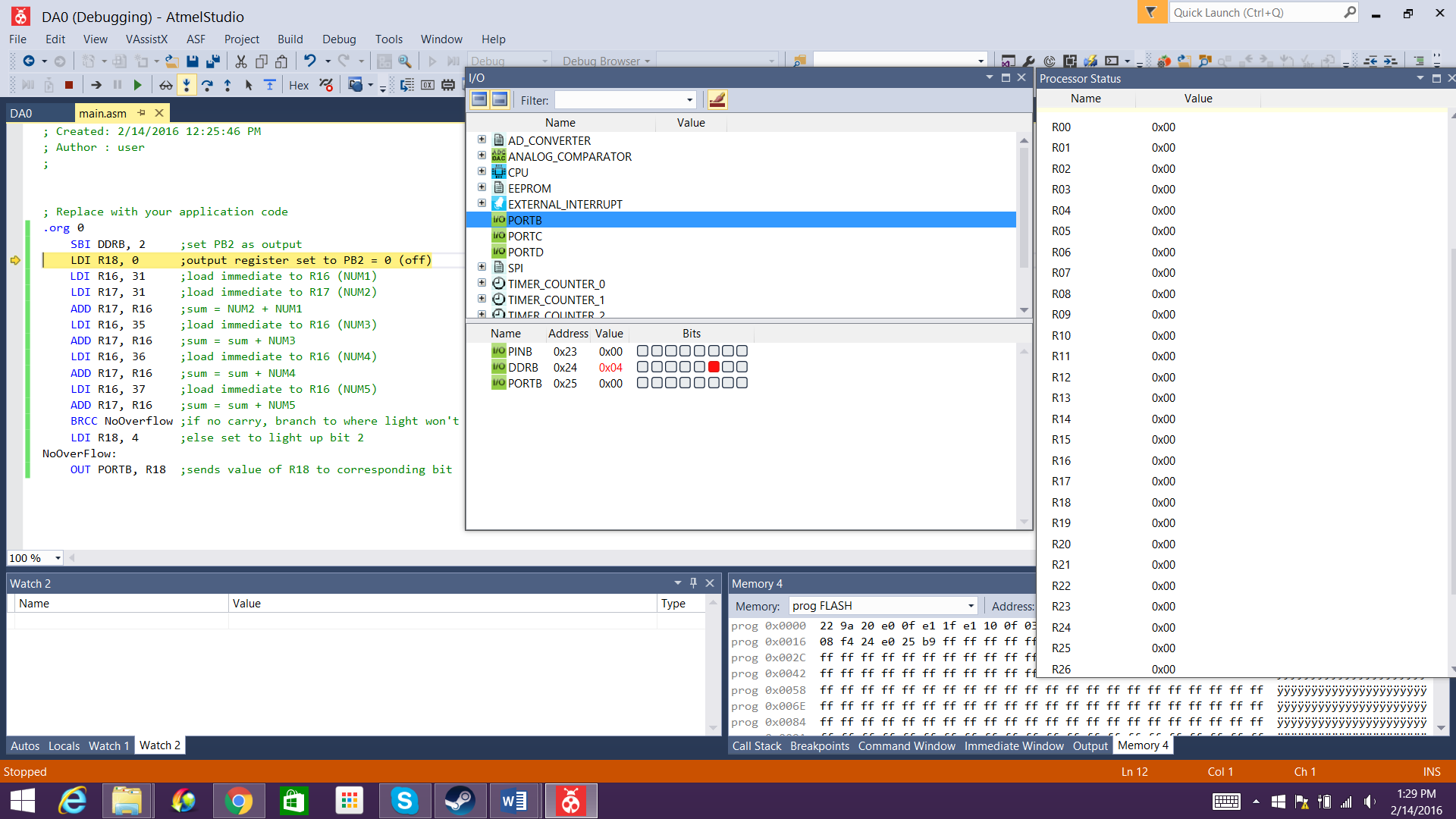
NoOverFlow:

OUT PORTB, R18 ;sends value of R18 to corresponding bit

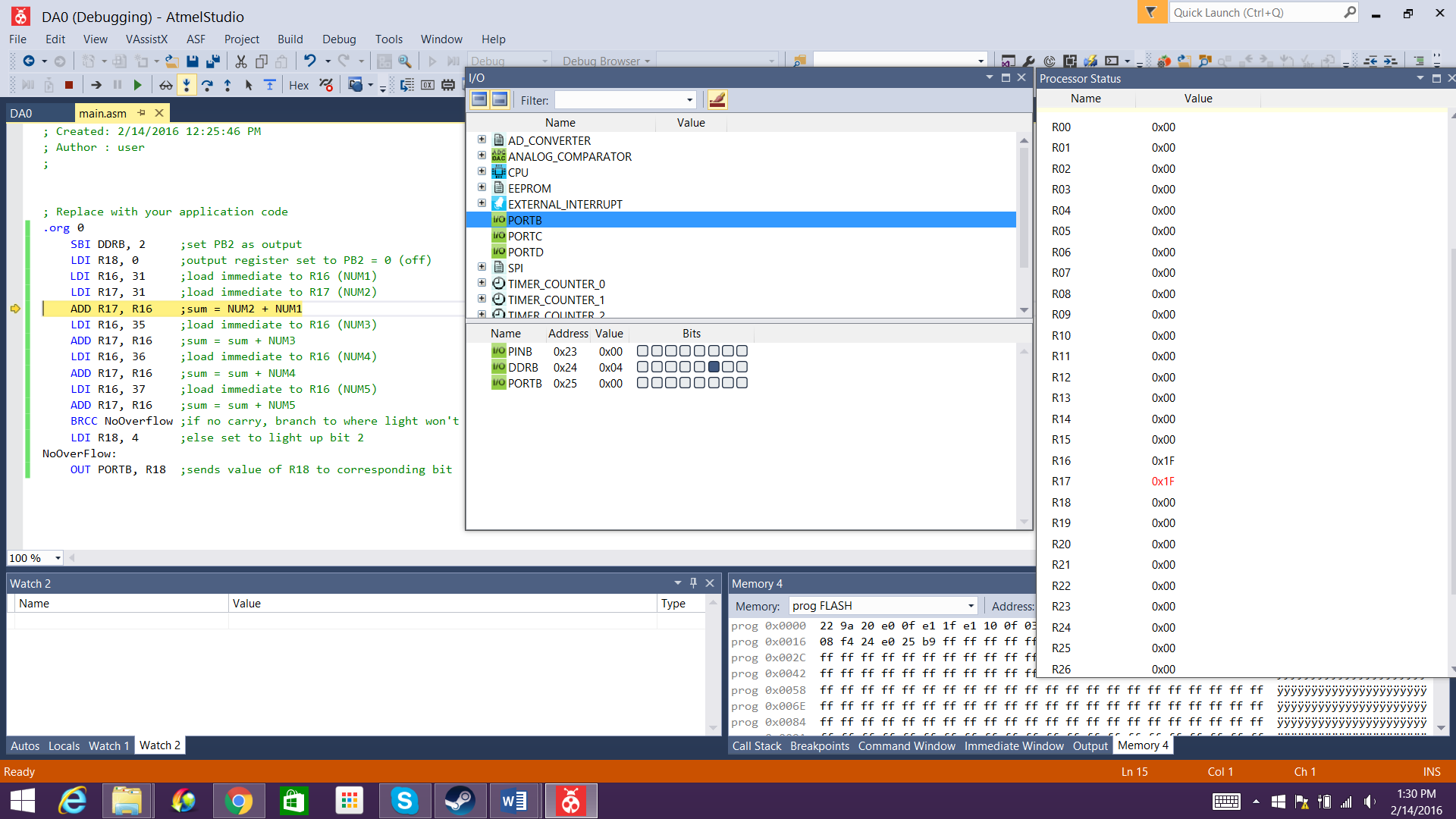
|  |  |  |  |
| --- | --- | --- | --- |
| 7. | SCREENSHOTS OF EACH TASK OUTPUT |  |  |

**TASK A:**

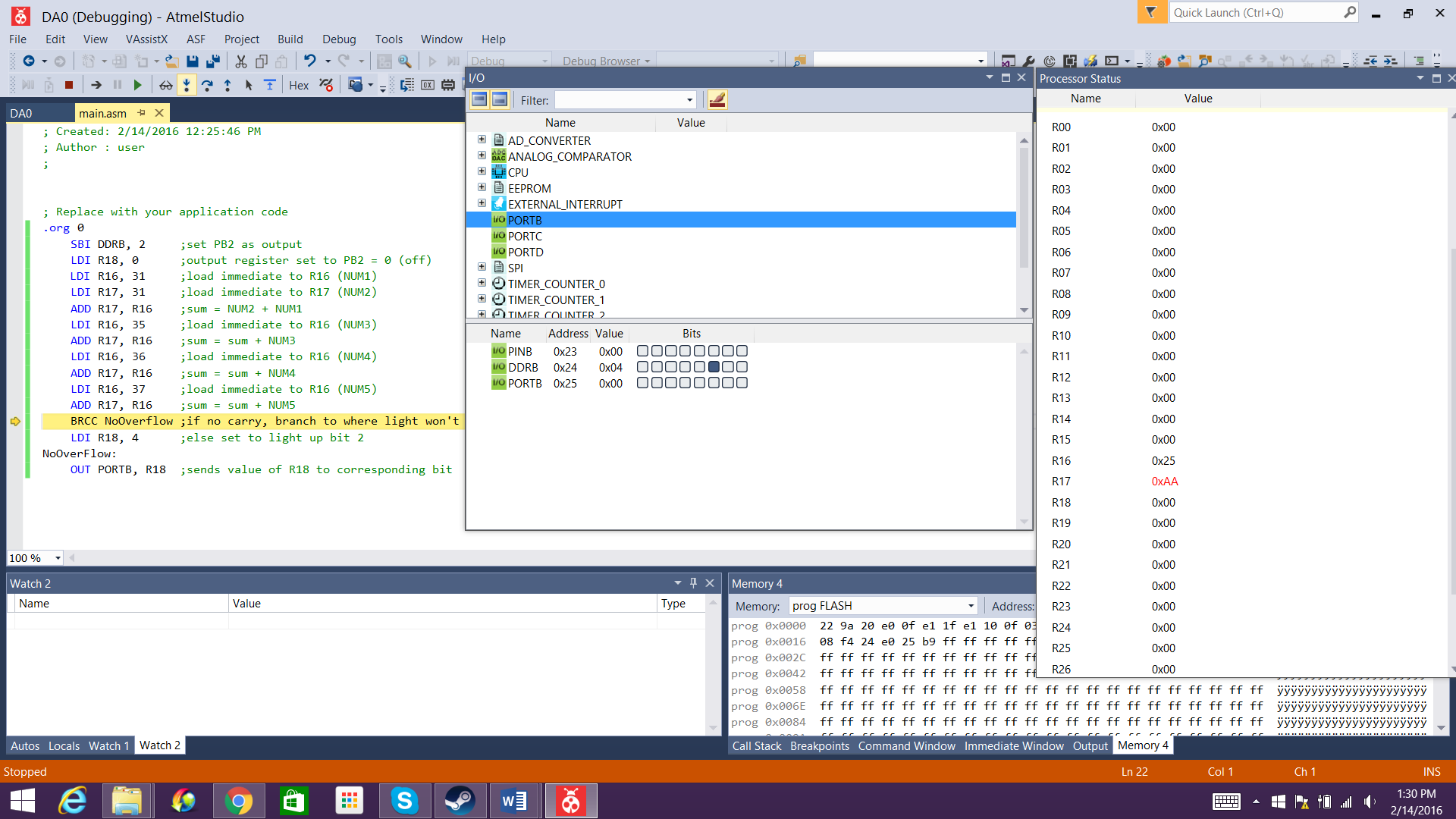
Sum = 0xAA (170)



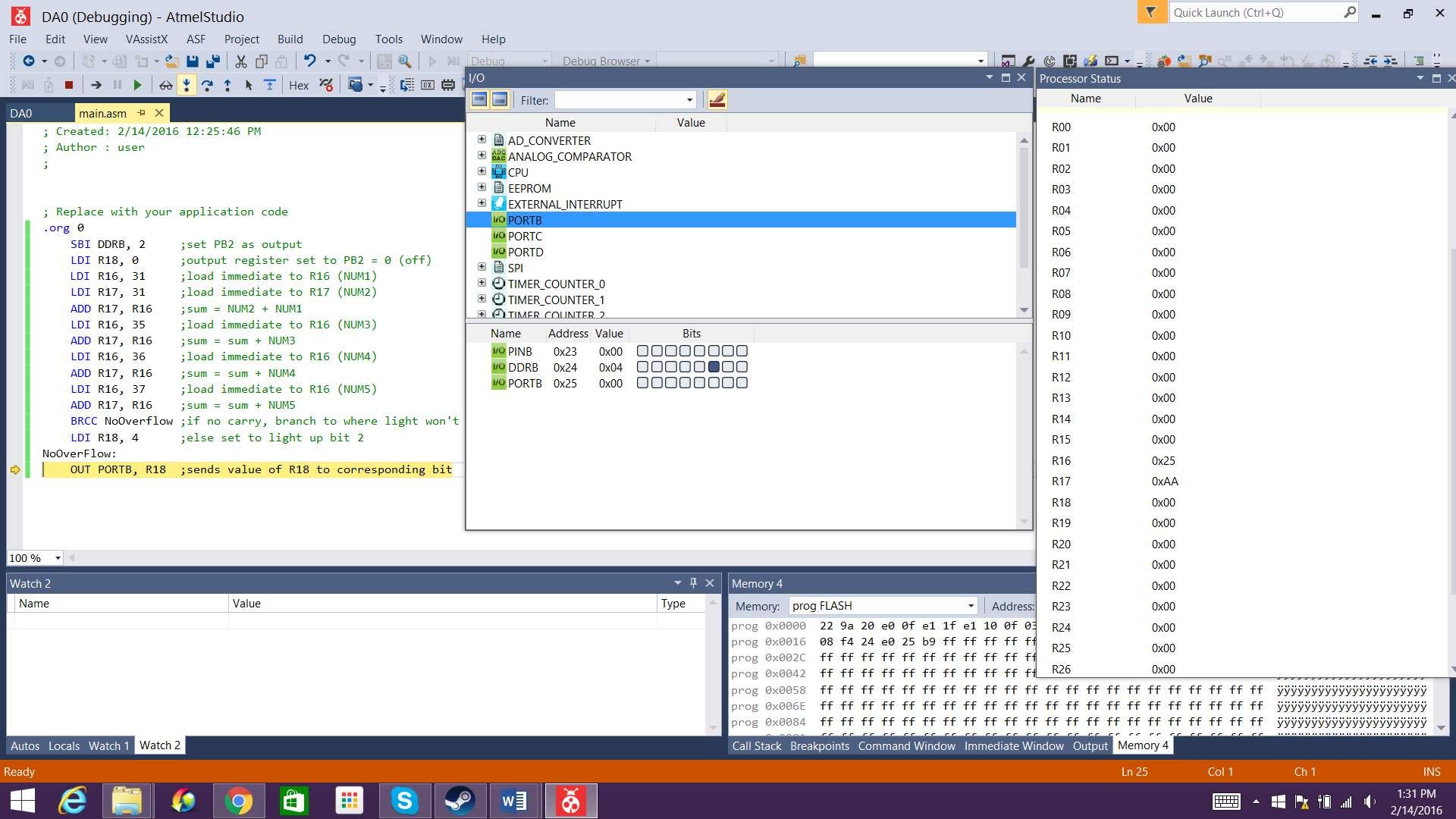
PB2 enabled as output. R18 set as 0 by default (PB2 off).



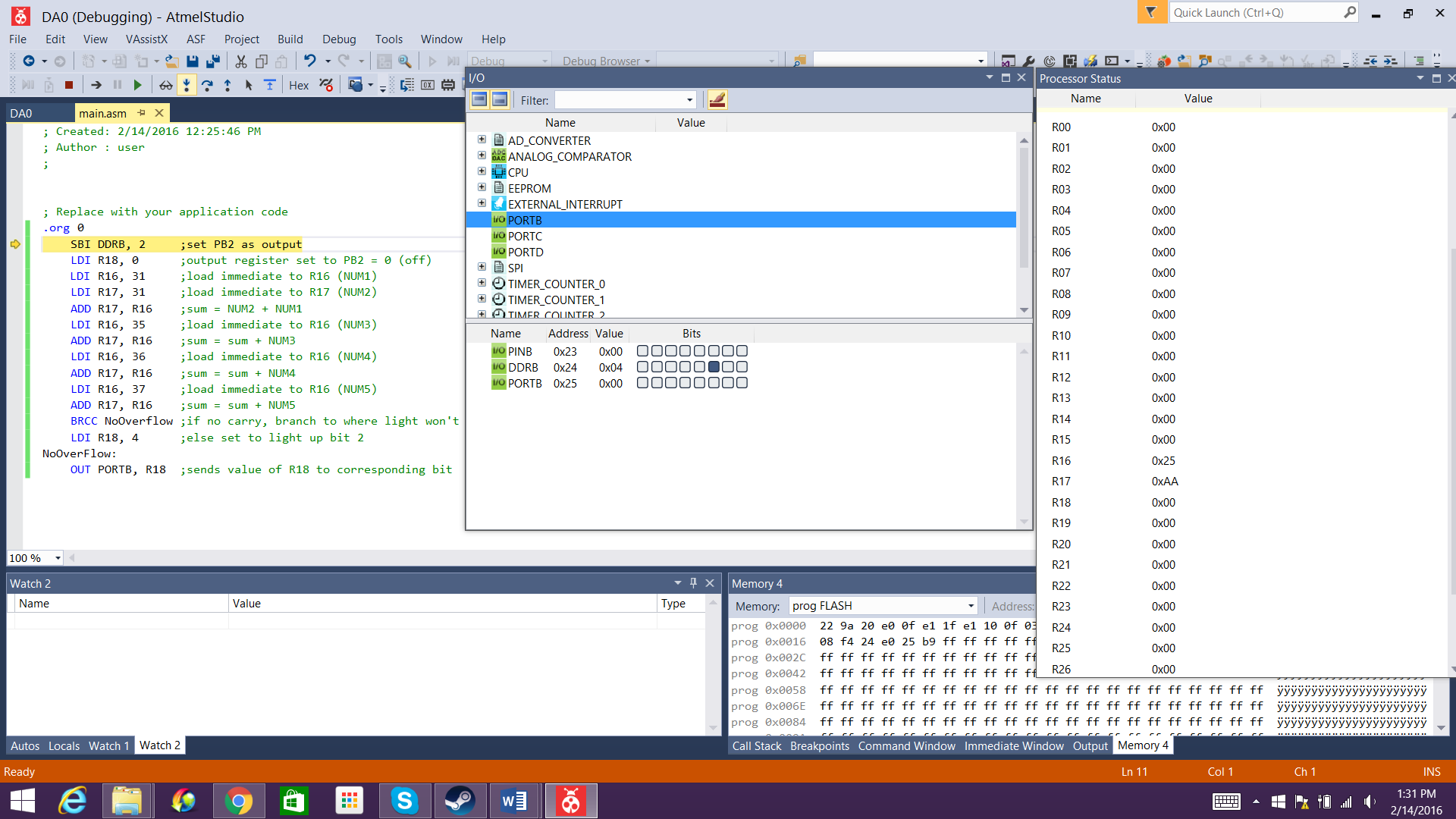
R16 and R17 loaded with 1st number and 2nd number, respectively.



Overall sum is computed and stored in R17. In this case, the sum is 0xAA.

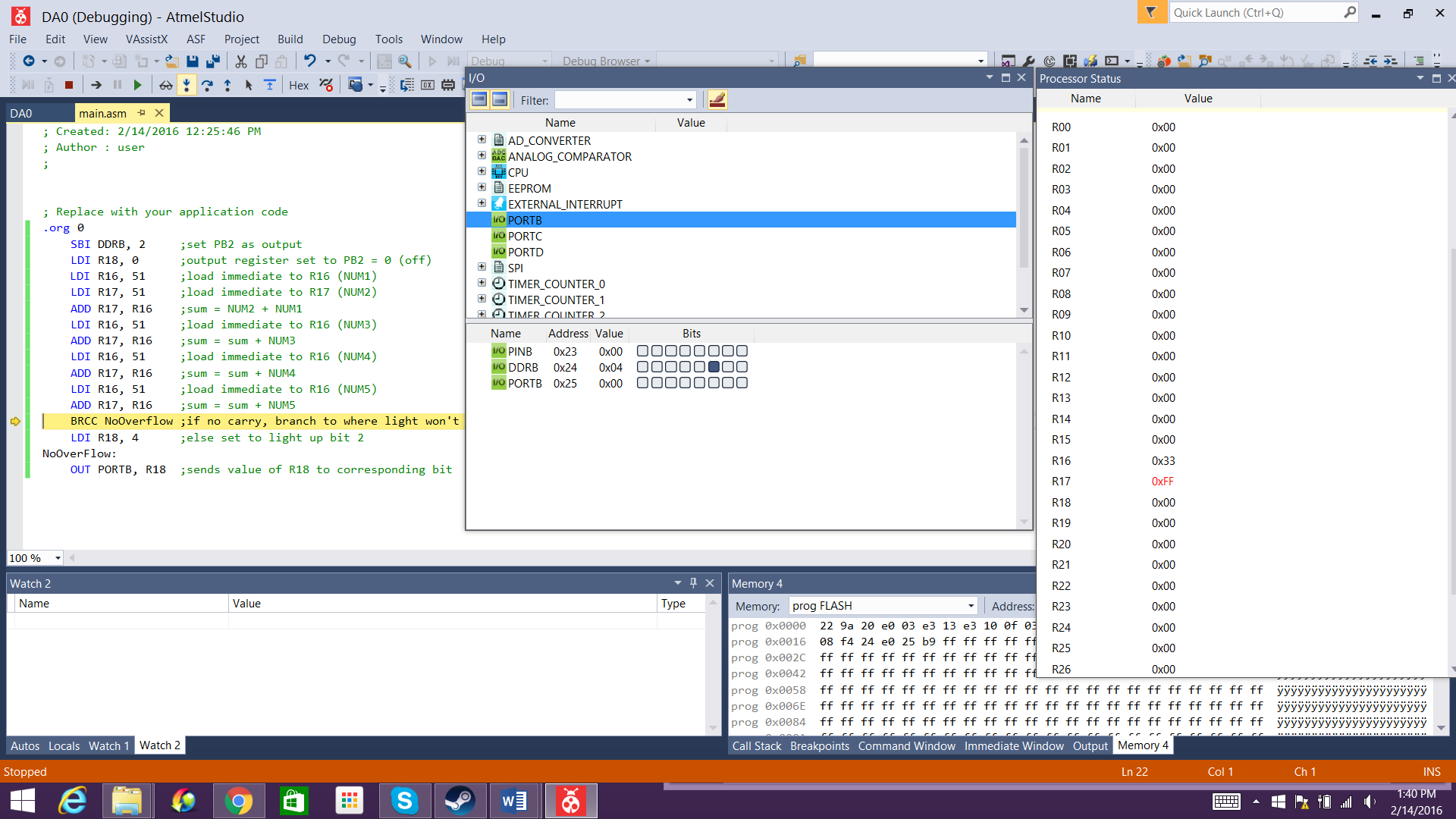


Since there is no overflow (sum remains at two hex digits), the program jumps to NoOverflow and R18 remains at 0.

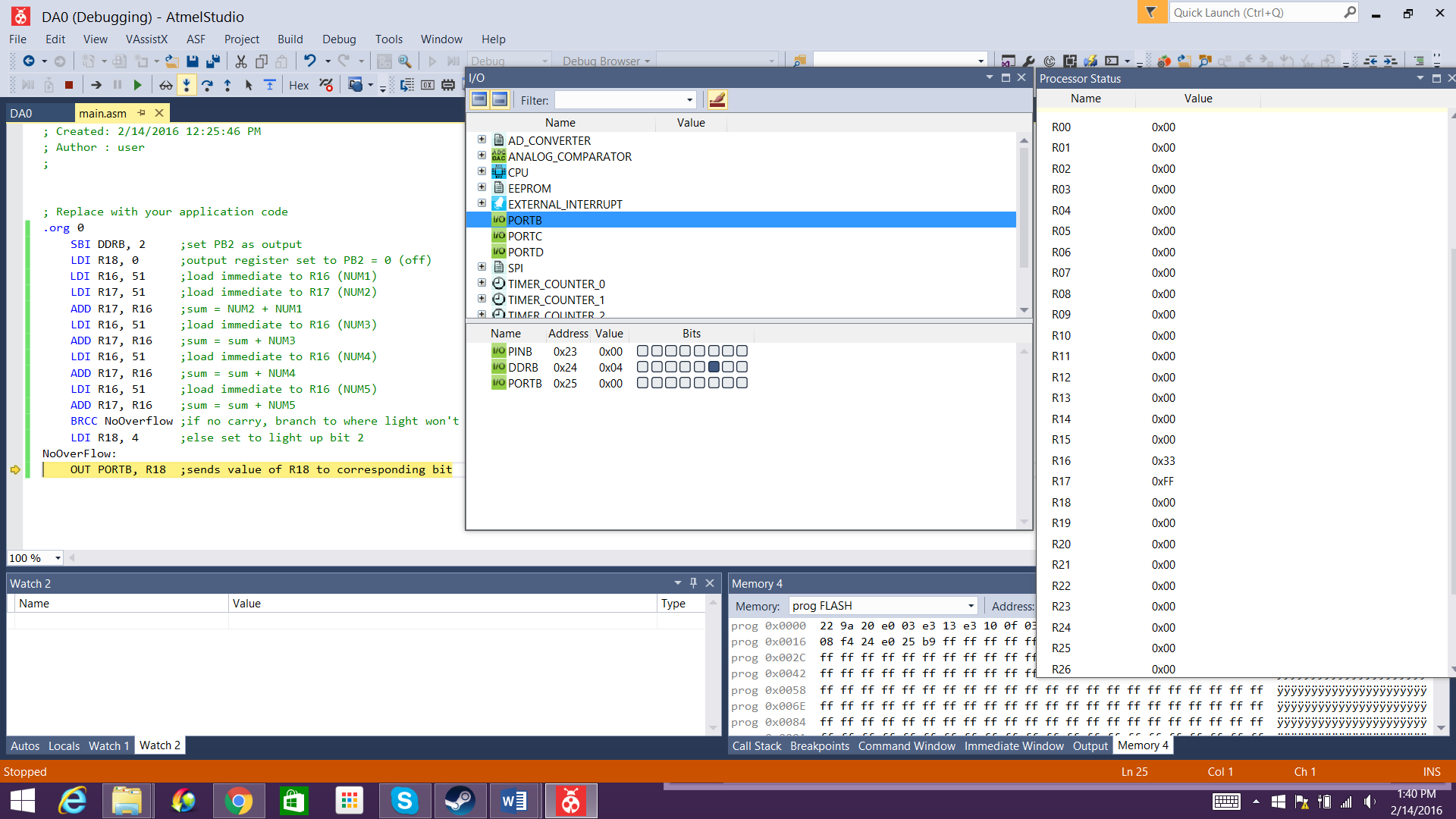


The value of R18 is outputted to PORTB. In this case it is 0, causing PB2 to remain off (low).

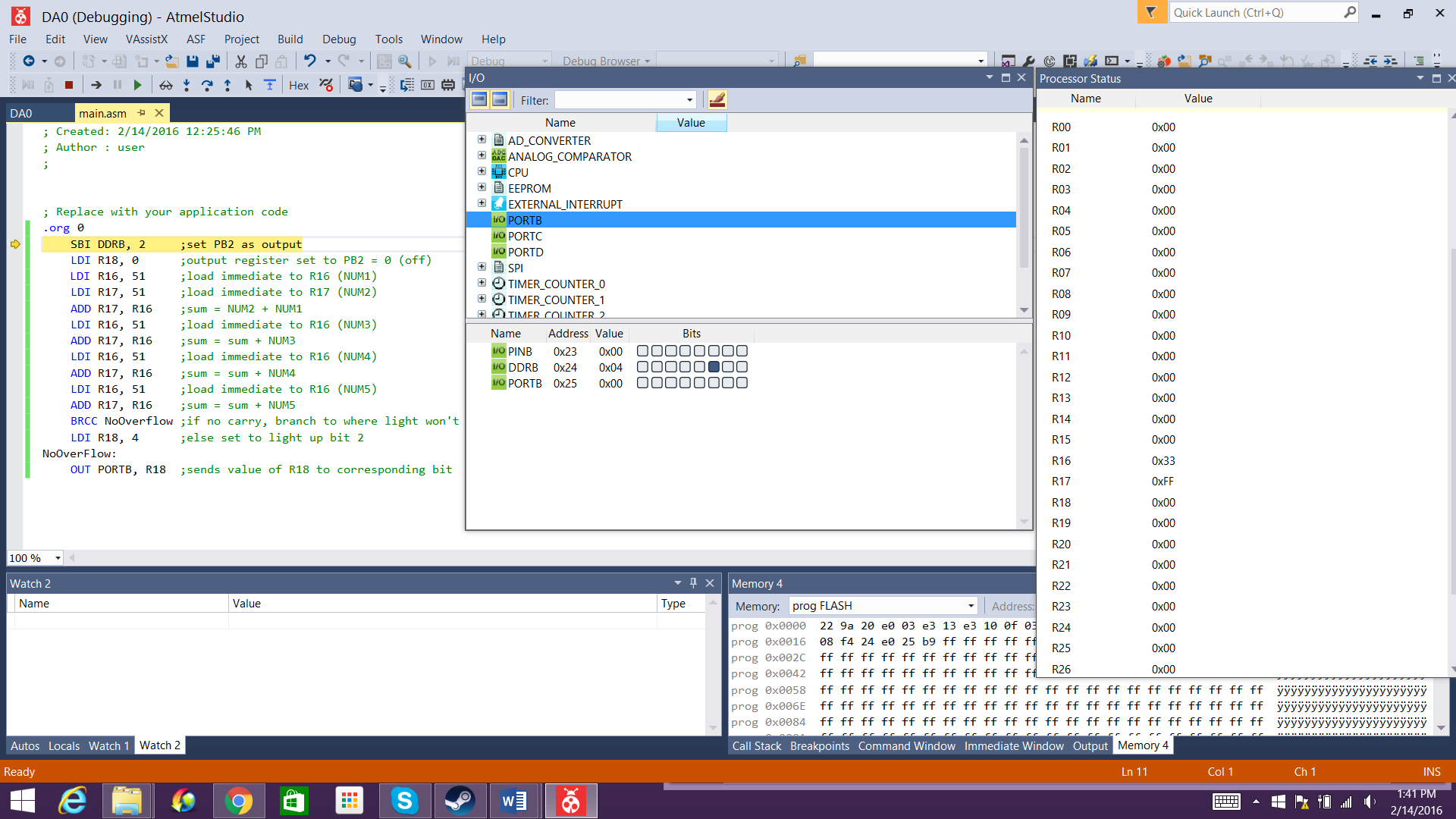
Sum = 0xFF (255)



Sum is computed and stored in R17. The sum is 0xFF.

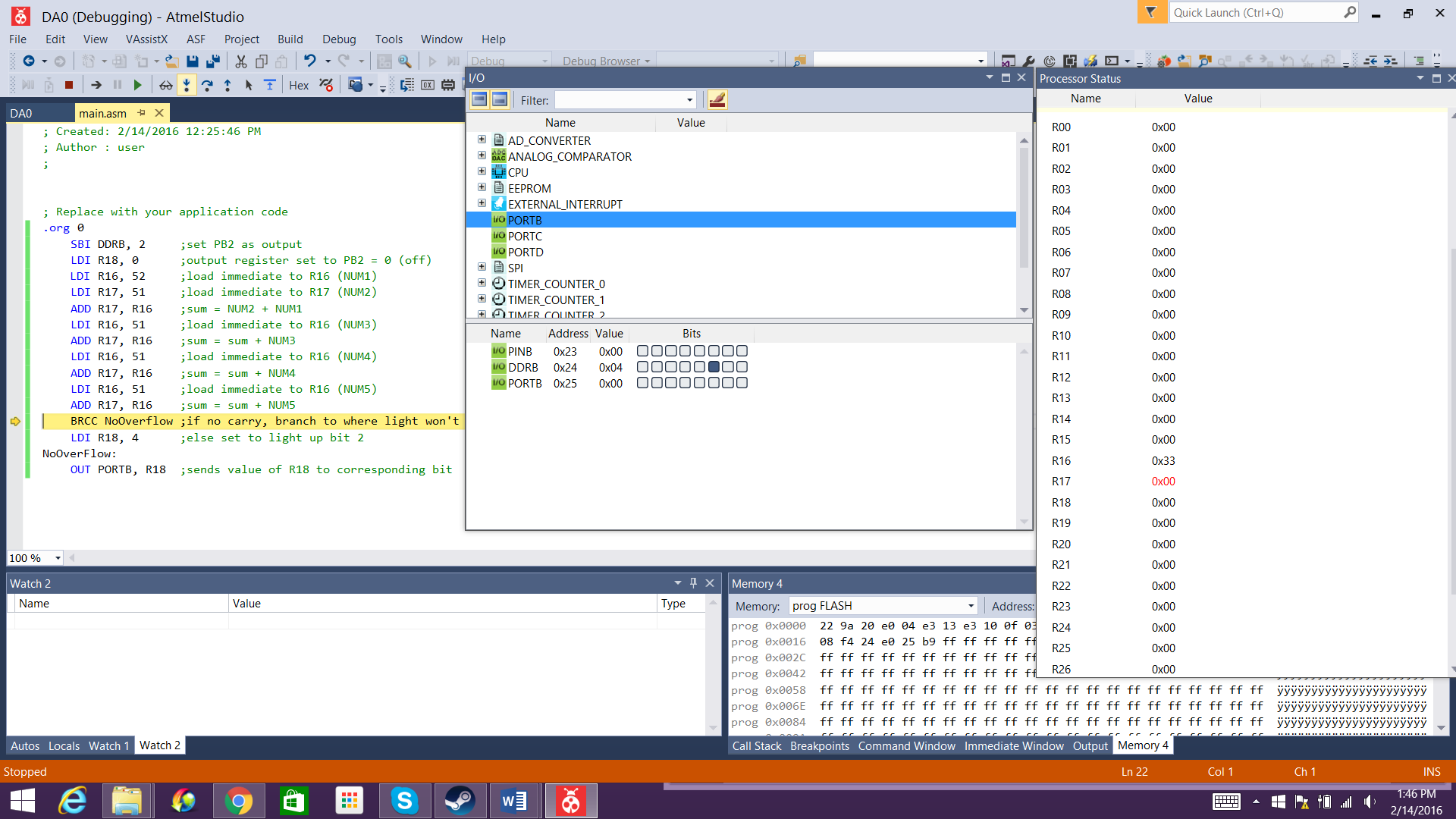


Since there is no overflow (at 2 hex digits), the program branches to NoOverflow and R18 remains at 0.

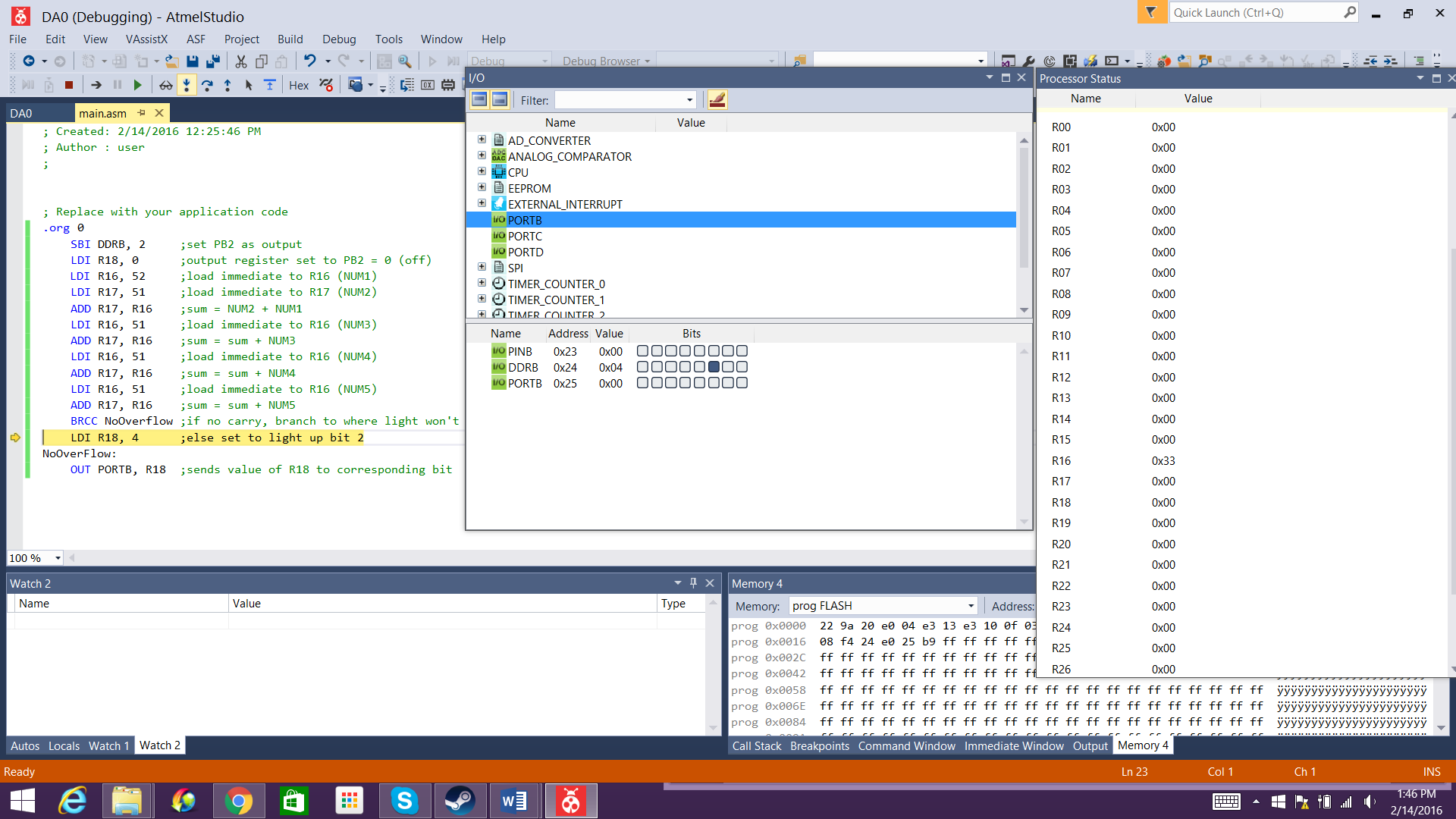


The value of R18 is outputted to PORTB. PB2 remains off due to the value being 0.

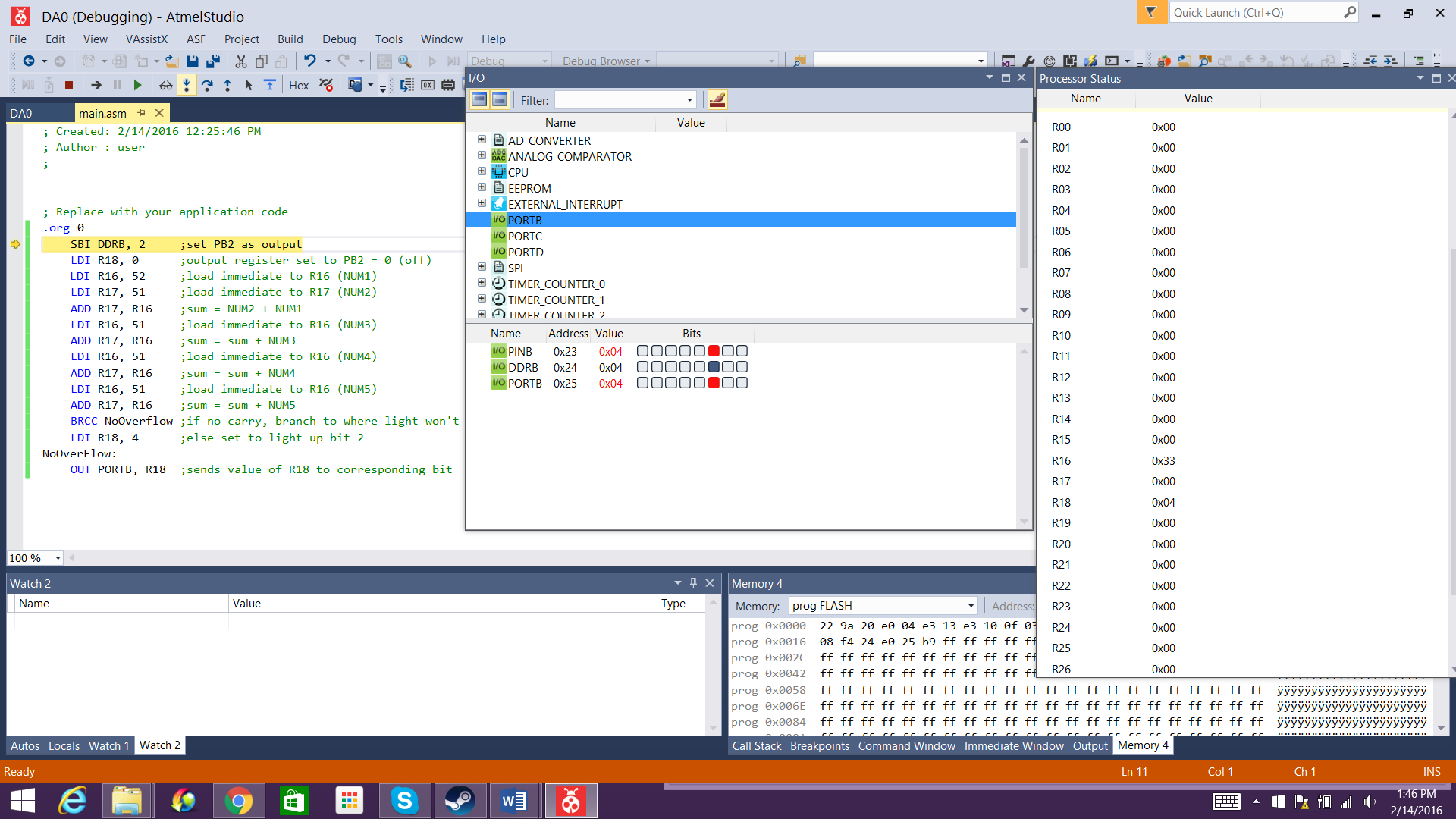
Sum = 0x100 (256)



The sum is computed and stored in R17. The sum is 0x100. Since each register is only 8 bits long, only the last two hex digits of this numbers are stored. Hence, R17 = 0x00.

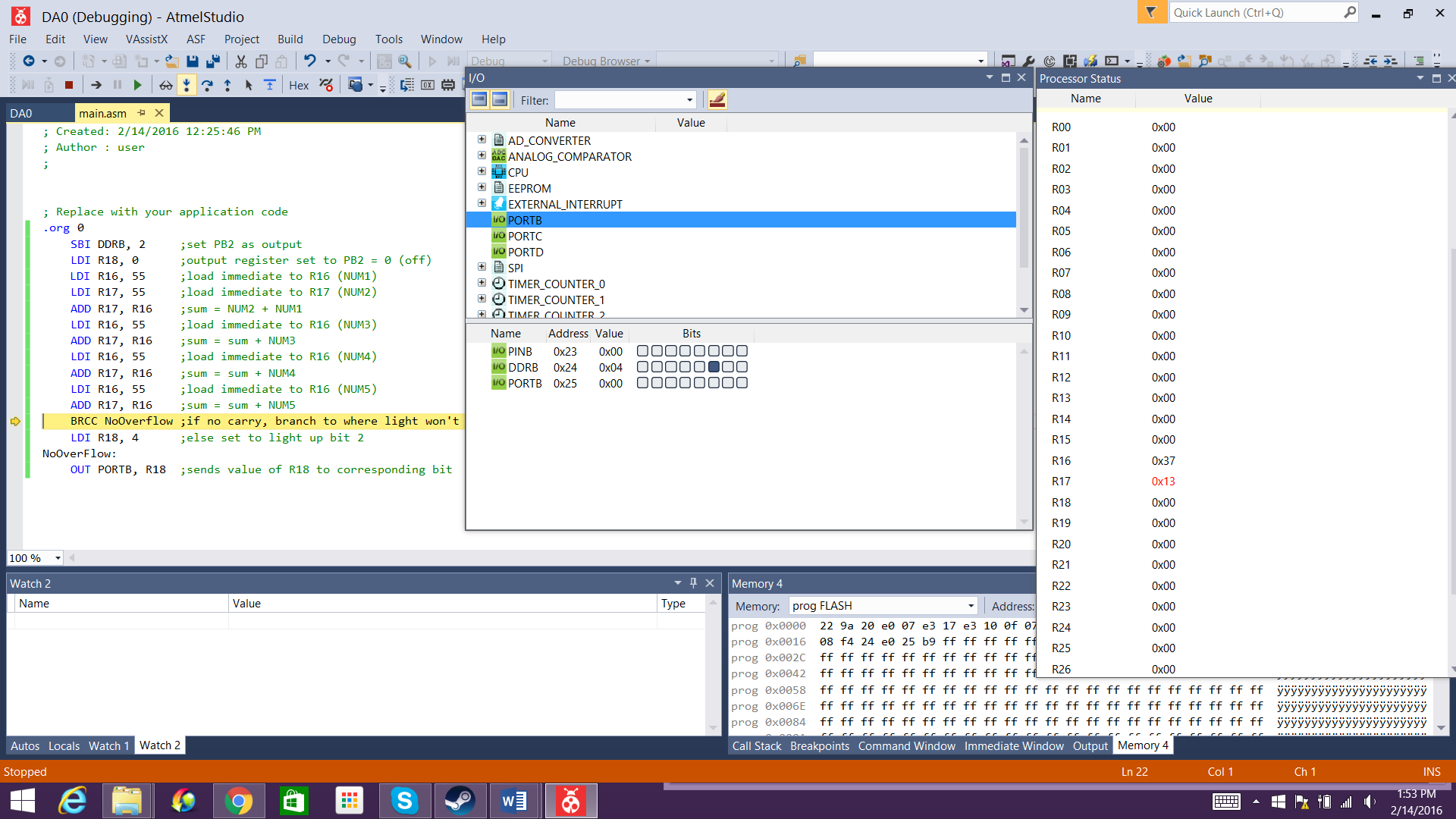


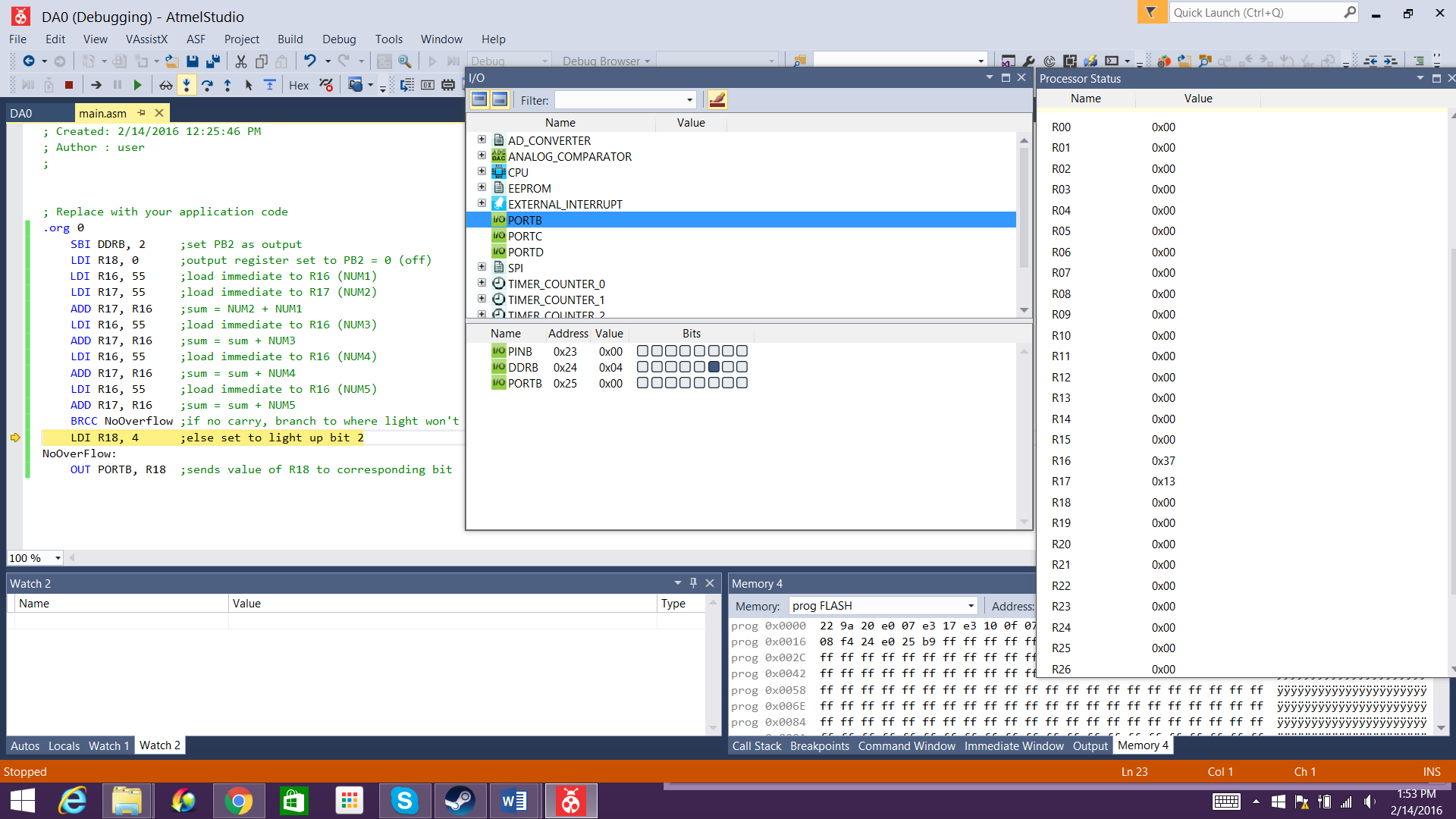
Since the branch instruction is BRCC, which means to branch to the designated address if there is no carry, the program does not jump to NoOverflow due to the fact that there is a carry in this case.

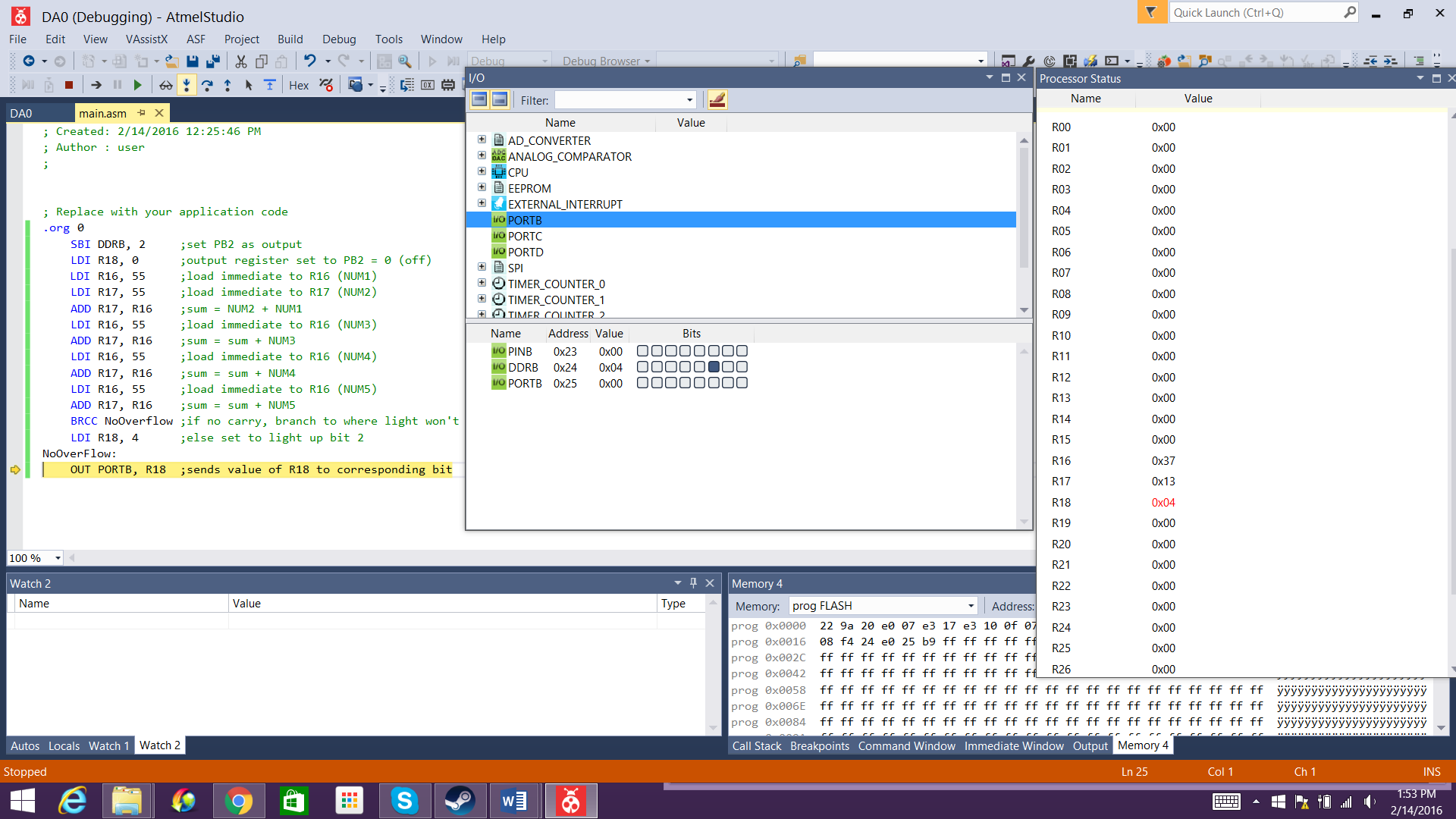


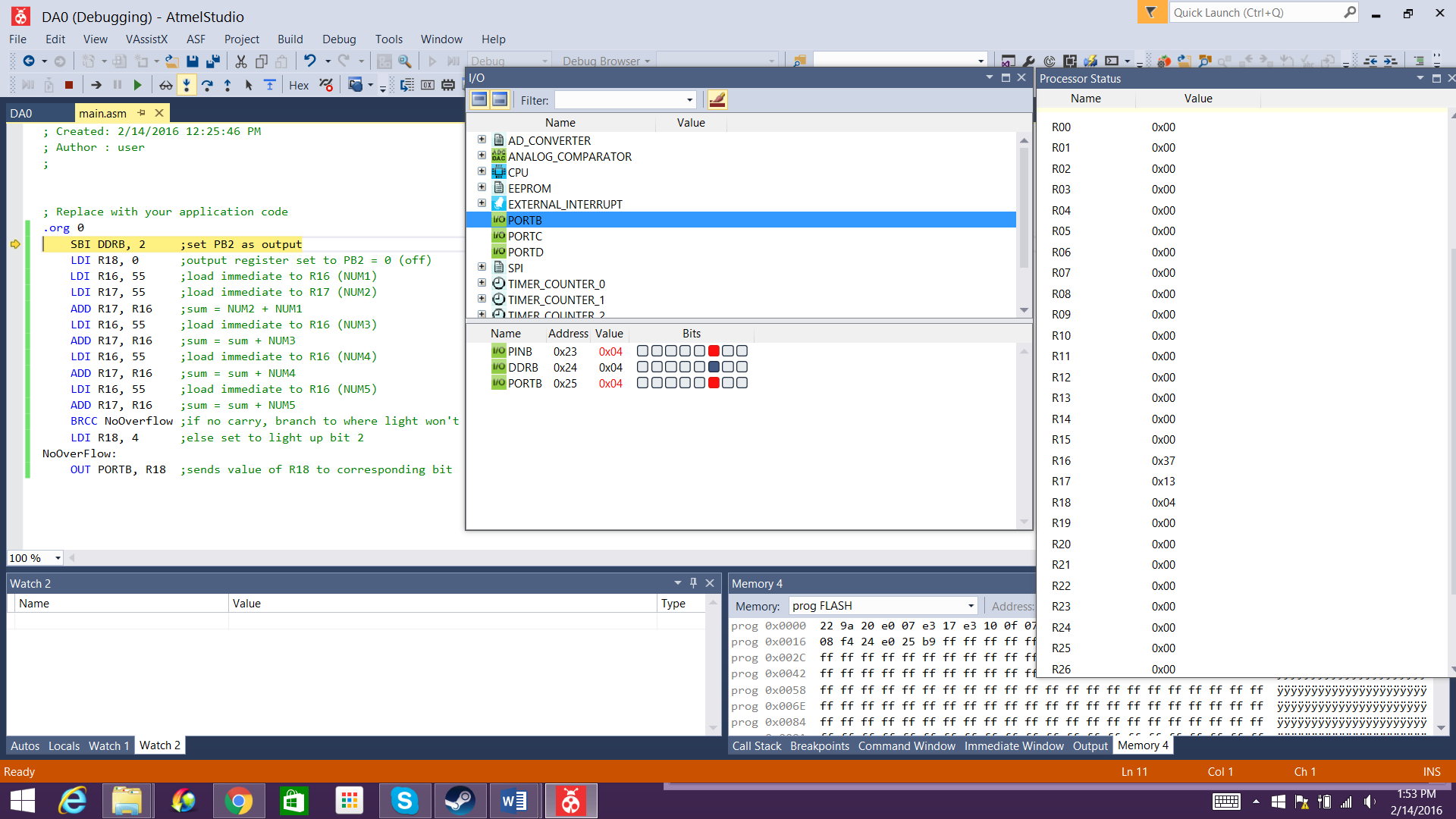
Consequently, R18 is set to 4 (0x04). When the value of R18 is outputted to PORTB, PB2 (the corresponding bit) lights up.

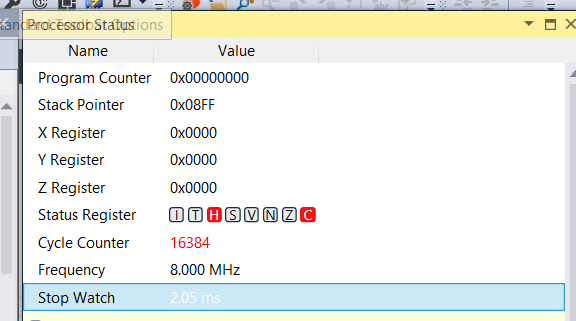
Sum = 0x113 (275)

As seen below, the same applies for an even larger value. PB2 lights up as expected.   










|  |  |  |  |
| --- | --- | --- | --- |
| 10. | GOOGLECODE LINK OF THE DA |  |  |
| http:// @svn or github repository link | | | |

**Student Academic Misconduct Policy**

<http://studentconduct.unlv.edu/misconduct/policy.html>

“This assignment submission is my own, original work”.

CANDACE CAIMOL