Trace Stewart

CPE301 – SPRING 2018

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)** |
| 1 | COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS |  |  |
| 2. | INITIAL CODE OF TASK 1/A |  |  |
| 3. | INCREMENTAL / DIFFERENTIAL CODE OF TASK 2/B |  |  |
| 3. | INCREMENTAL / DIFFERENTIAL CODE OF TASK 3/C |  |  |
| 3. | INCREMENTAL / DIFFERENTIAL CODE OF TASK 4/D |  |  |
| 3. | INCREMENTAL / DIFFERENTIAL CODE OF TASK 5/E |  |  |
| 4. | SCHEMATICS |  |  |
| 5. | SCREENSHOTS OF EACH TASK OUTPUT |  |  |
| 5. | SCREENSHOT OF EACH DEMO |  |  |
| 6. | VIDEO LINKS OF EACH DEMO |  |  |
| 7. | GOOGLECODE LINK OF THE DA |  |  |
|  |  |  |  |
|  |  |  |  |

1. **COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS**

None.

1. **INITIAL/DEVELOPED CODE OF TASK 1/A**

RESET:

.equ STARTADDS = 0x0222 ; Starting Address

.equ count = 255 ; Part 1 of the counter to 300

.equ count1 = 46 ; Part 2 of the counter to 300

.org 0

clr r0 ; Clearing r0 register

ldi XL, low(STARTADDS) ; XL = STARTADDS[7:0]

ldi XH, high(STARTADDS) ; XH = STARTADDS[15:8]

ldi YL, low(0x400) ; YL = 0x00

ldi YH, high(0x400) ; YH = 0x04

ldi ZL, low(0x600) ; ZL = 0x00

ldi ZH, high(0x600) ; ZH = 0x06

; Initializing count registers

ldi r21, count ; r21 = 255

ldi r22, count1 ; r22 = 46

ldi r20, 5 ; r20 = 5

; Clearing Registers for sums

clr r0 ; R0 = 0

clr r16 ; R16 = 0

clr r17 ; R17 = 0

clr r18 ; R18 = 0

clr r19 ; R19 = 0

clr r5 ; r5 = 0

inc r5 ; r5 = 1

START:

; Adding lower and upper address bits to be stored

mov r1, XL ; R1 = XL

add r1, XH ; R1 = XL + XH

mov r3, r1 ; R3 = R1

st X+, r3 ; [X] = XL + XH

DONE:

cp r21, r0 ; if(count == 0)

breq NOTYET ; go to NOTYET

sub r21, r5 ; R21 = R21 = R5

brne START ; if(R21 != 0) go to START

NOTYET:

sub r22, r5 ; R22 = R22 - R5

brne Start ; if(R22 != 0) go to START

FIN:

rjmp RESET ; Reset

**MODIFIED CODE OF TASK 2/A from TASK 1/A**

; Checking if divisible by 5

DIVBYFIVE:

cp r1, r20 ; if(R1 < 5)

brlo DIVBAD ; go to DIVBAD

sub r1, r20 ; R1 = R1 - R20

cp r1, r0 ; if(R1 == 0)

breq DIVGOOD ;go to DIVGOOD

rjmp DIVBYFIVE ; go to DIVBYFIVE

DIVGOOD:

st Z+, r3 ; [Z] = XL + XH

add r16, r3 ; R16 = R16 + R3 (lower)

adc r17, r0 ; R17 = R17 + R0 + C (upper)

rjmp DONE ; go to DONE

DIVBAD:

st Y+, r3 ; [Y] = XL + XH

add r18, r3 ; R18 = R18 + R3 (lower)

adc r19, r0 ; R19 = R19 + R0 + C (upper)

rjmp DONE ; goto DONE

DONE:

cp r21, r0 ; if(count == 0)

breq NOTYET ; go to NOTYET

sub r21, r5 ; R21 = R21 = R5

brne START ; if(R21 != 0) go to START

NOTYET:

sub r22, r5 ; R22 = R22 - R5

brne Start ; if(R22 != 0) go to START

FIN:

rjmp RESET ; Reset

1. **SCHEMATICS**

None

1. **SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)**

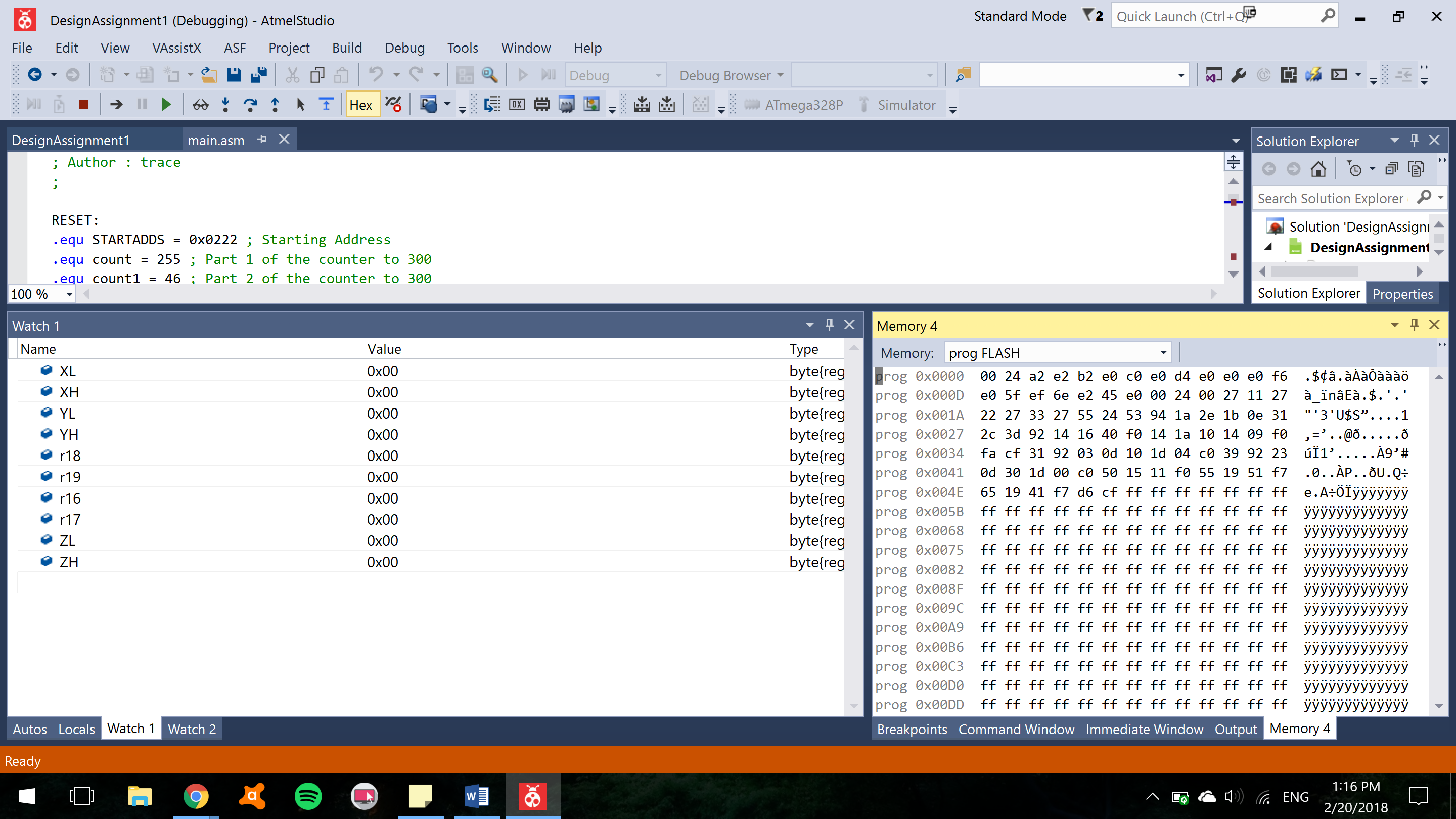


Figure Values of registers before program is started

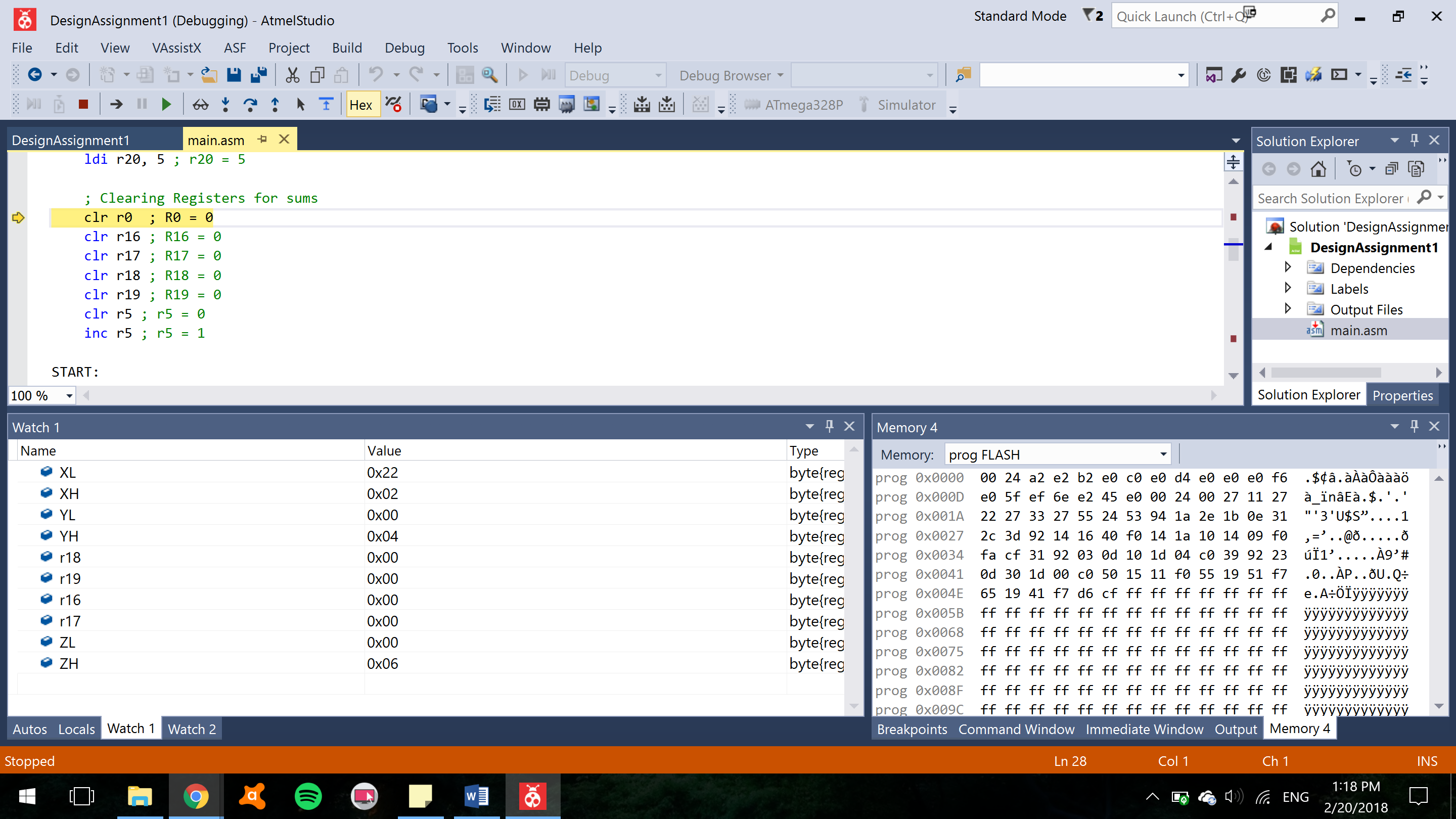


Figure Values of registers after addresses for pointers are stored in X,Y, and Z registers

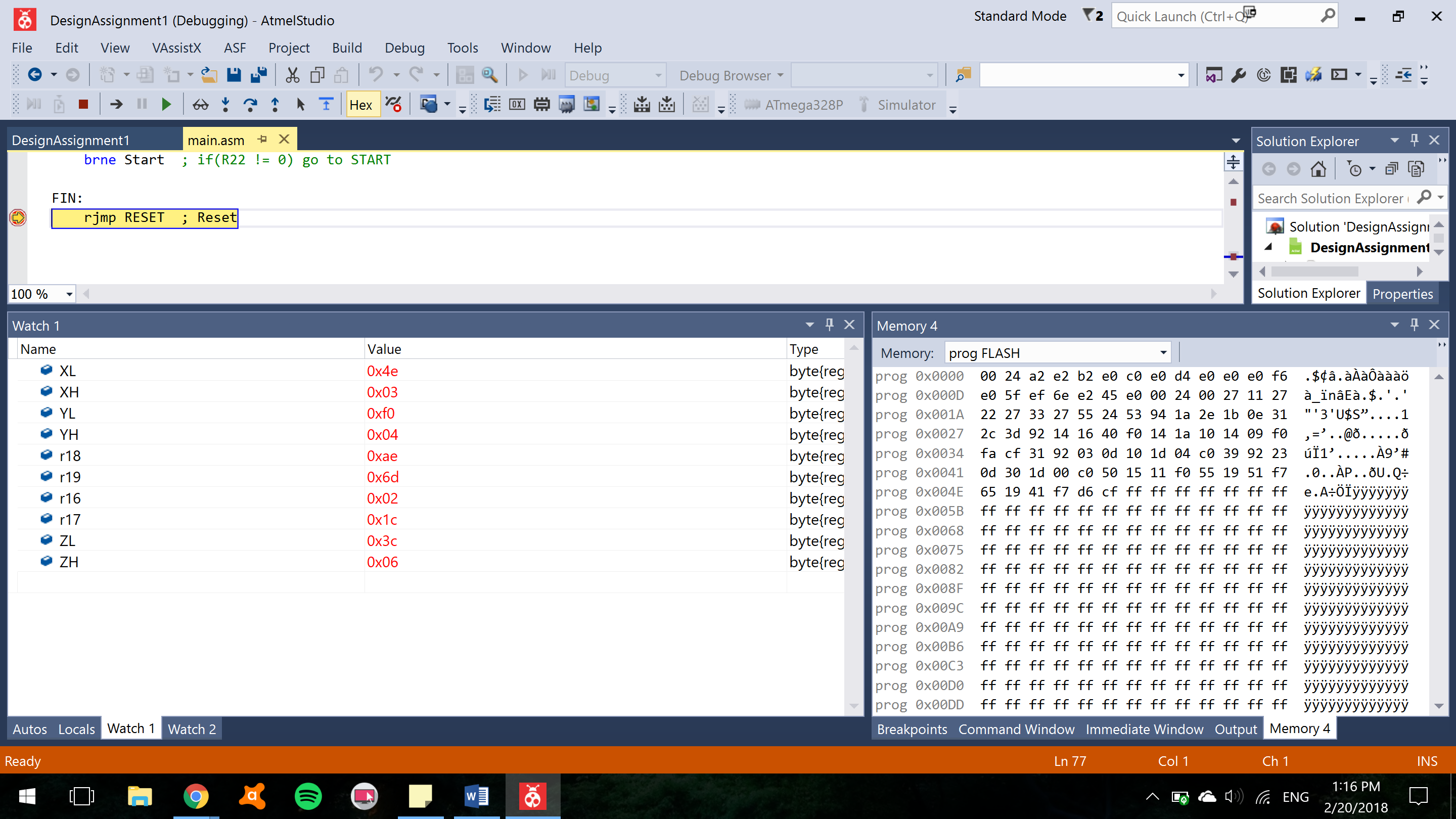


Figure Registers after the program has ran completely through. These are displayed in Hex values.

R16:R17 = **0x1C02** (Hex) = 7170 (Decimal)

R18:R19 = **0x6DAE** (Hex) = 28078 (Decimal)

X (Before) = 0x0222 (Hex) = 546 (Decimal)

X (After) = 0x034E (Hex) = 846 (Decimal)

Y (Before) = 0x0400 (Hex) = 1024 (Decimal)

Y (After) = 0x04F0 (Hex) = 1264 (Decimal)

Z (Before) = 0x0600 (Hex) = 1536 (Decimal)

Z (After) = 0x063C (Hex) = 1596 (Decimal)

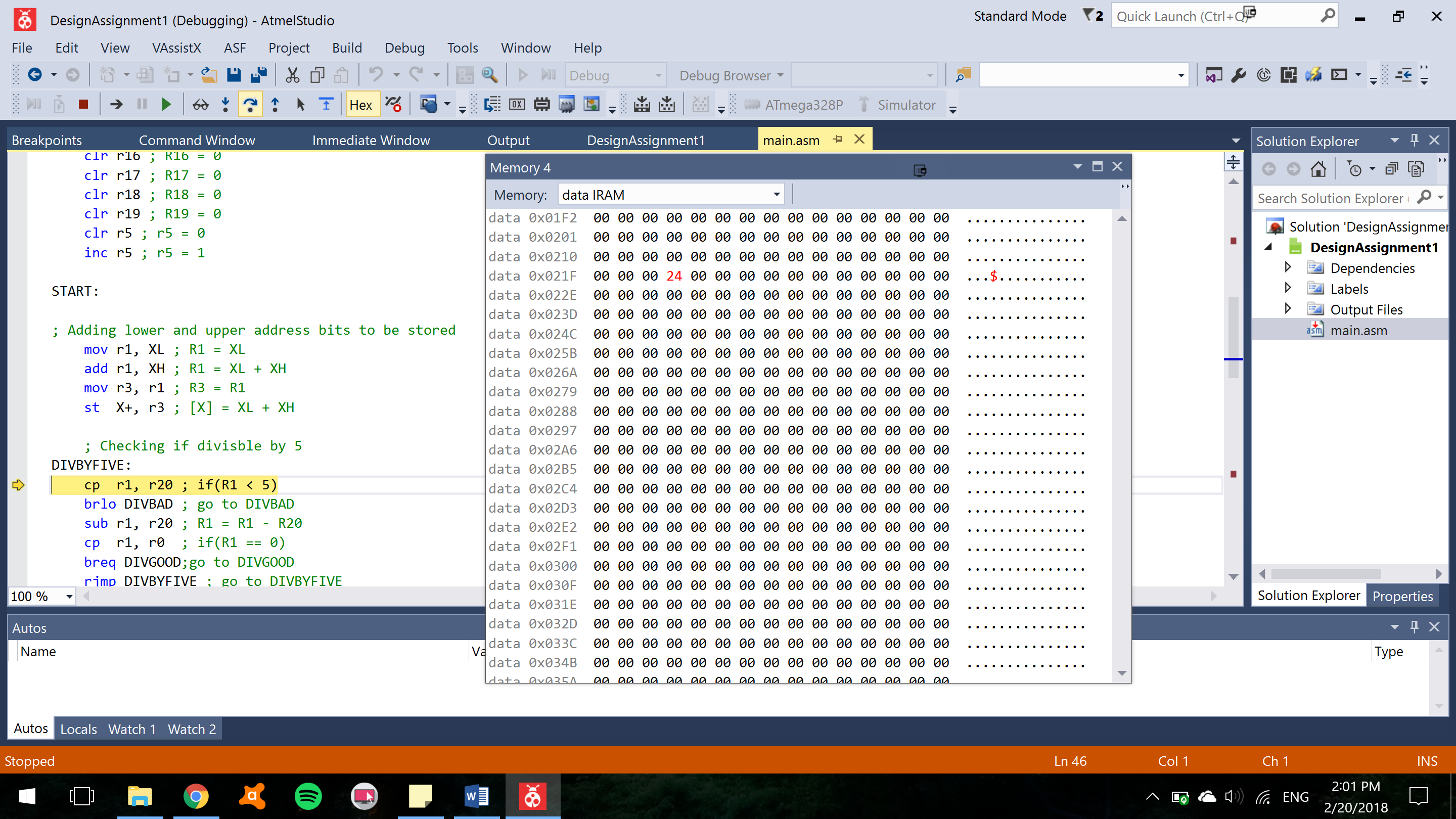


Figure Before storing values into any of the addresses

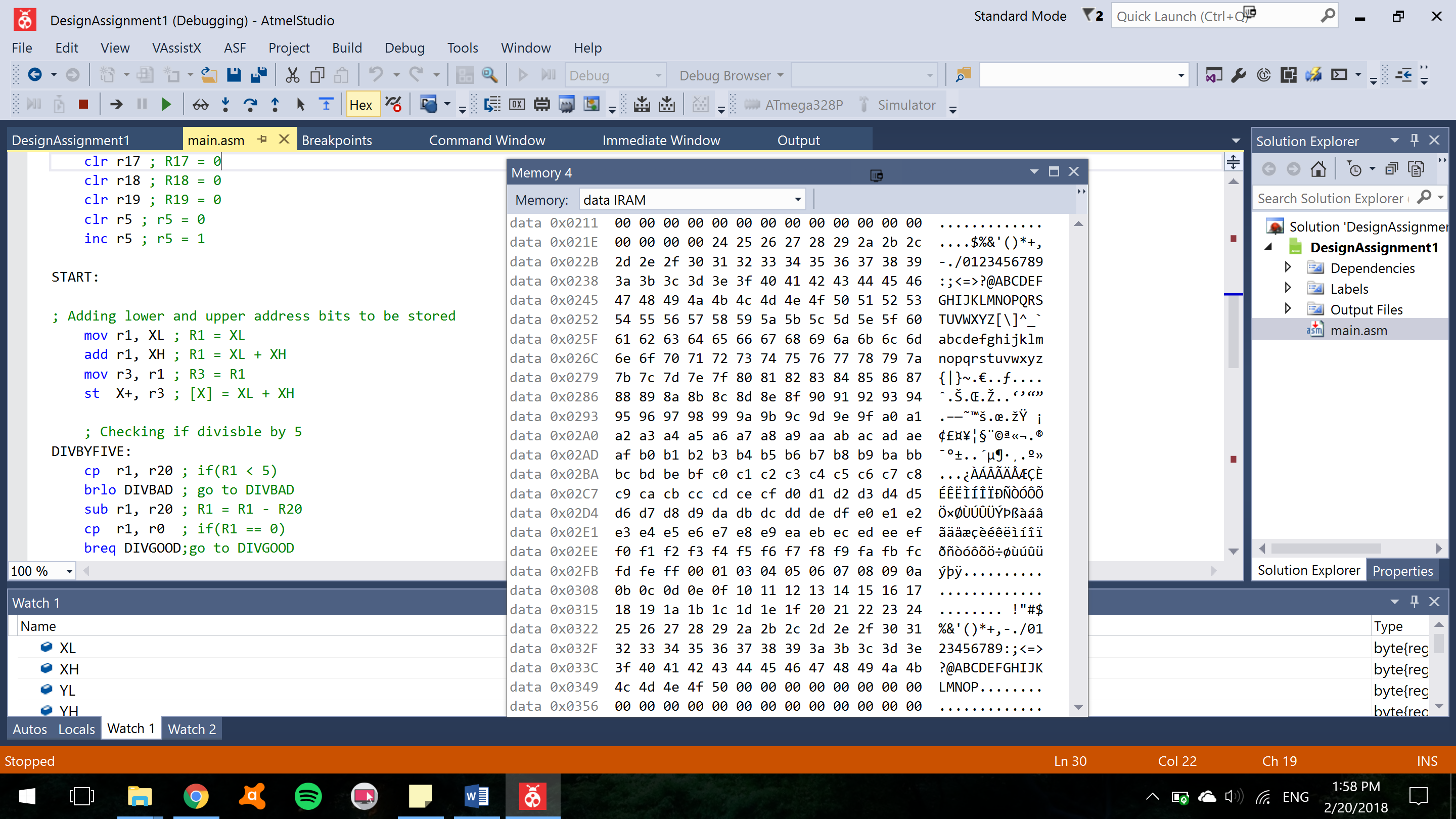


Figure After storing values from location 0x0222

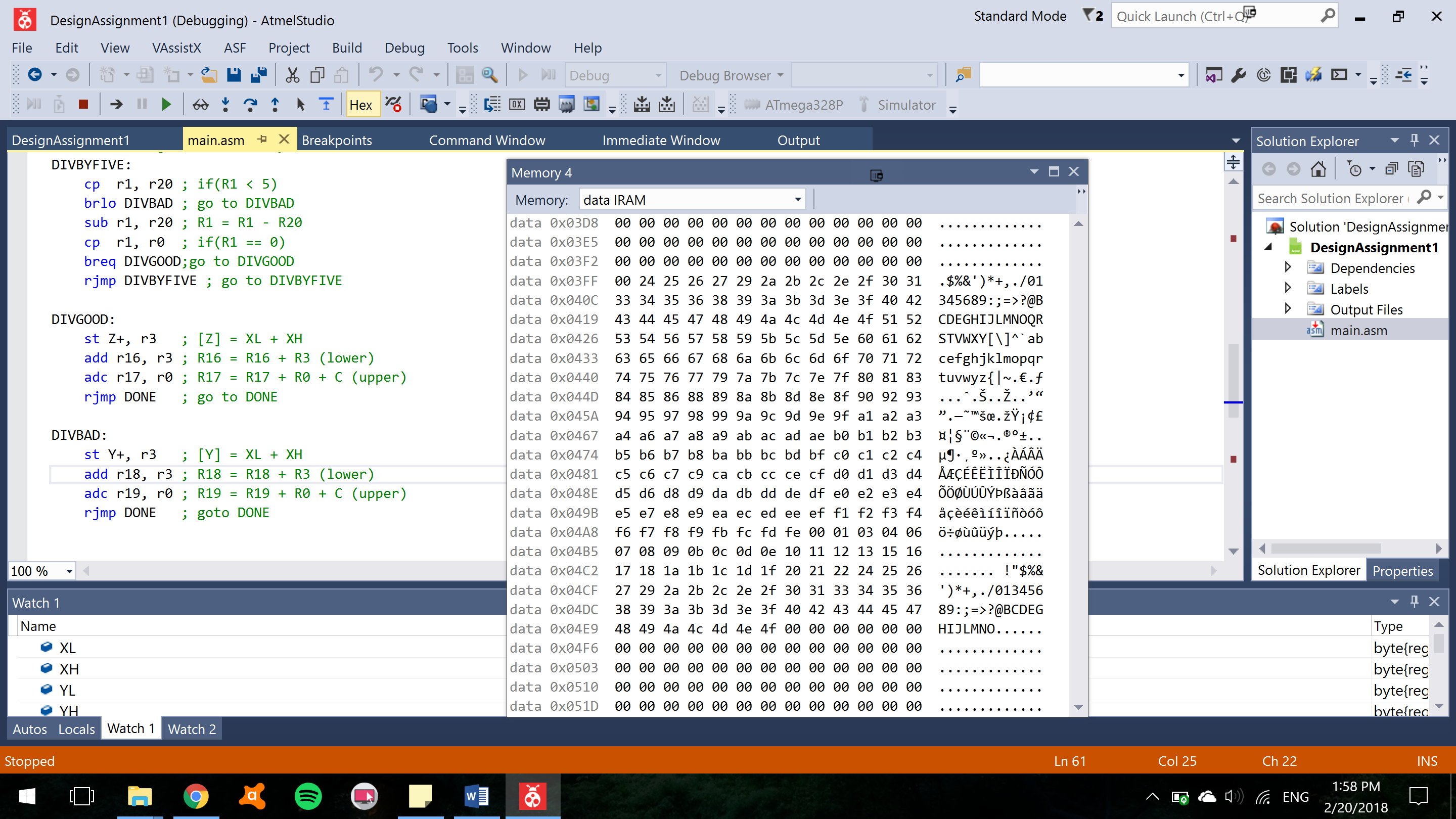


Figure After storing values from location 0x0400

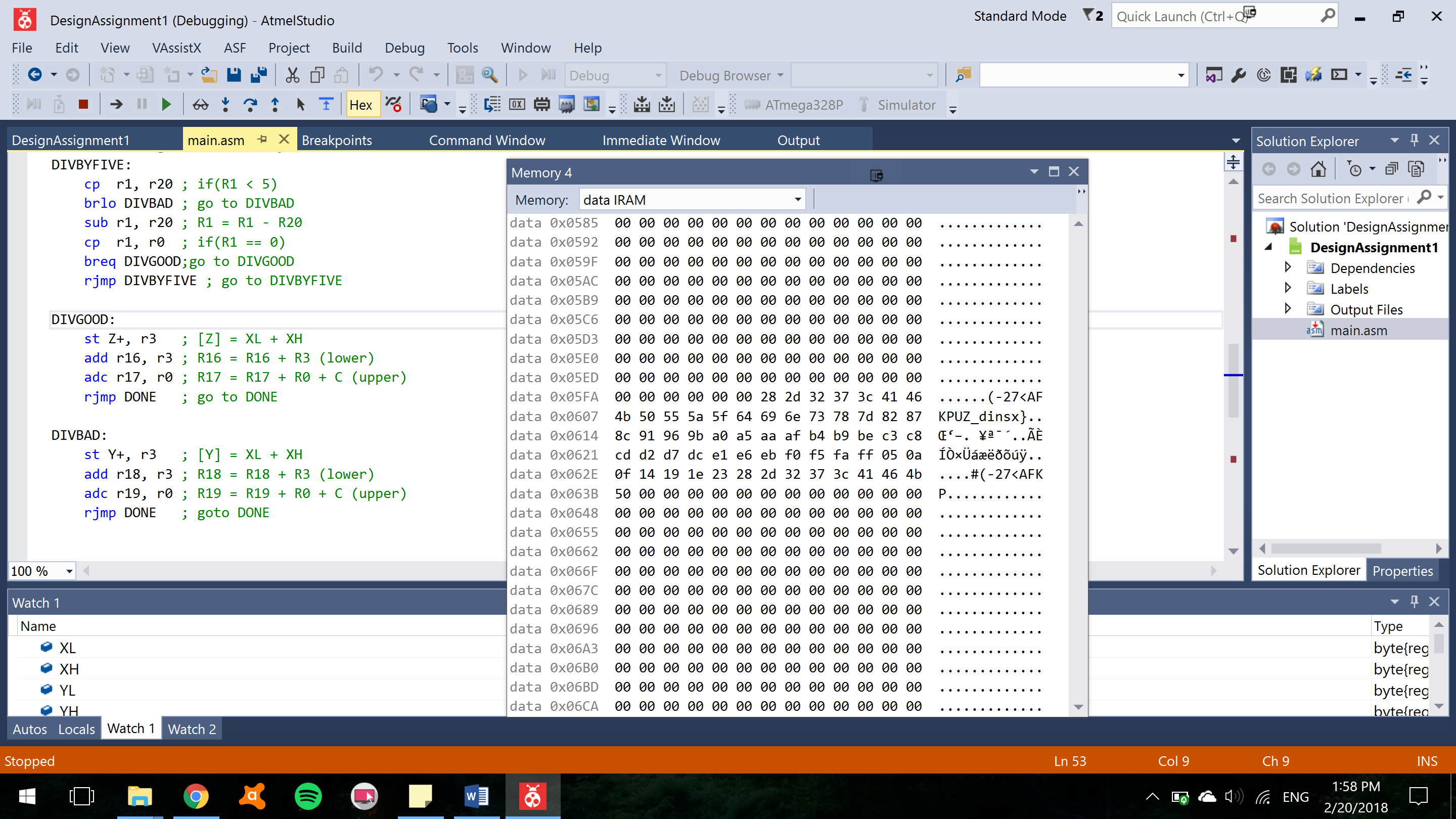


Figure After storing values from location 0x0600

TASK 1

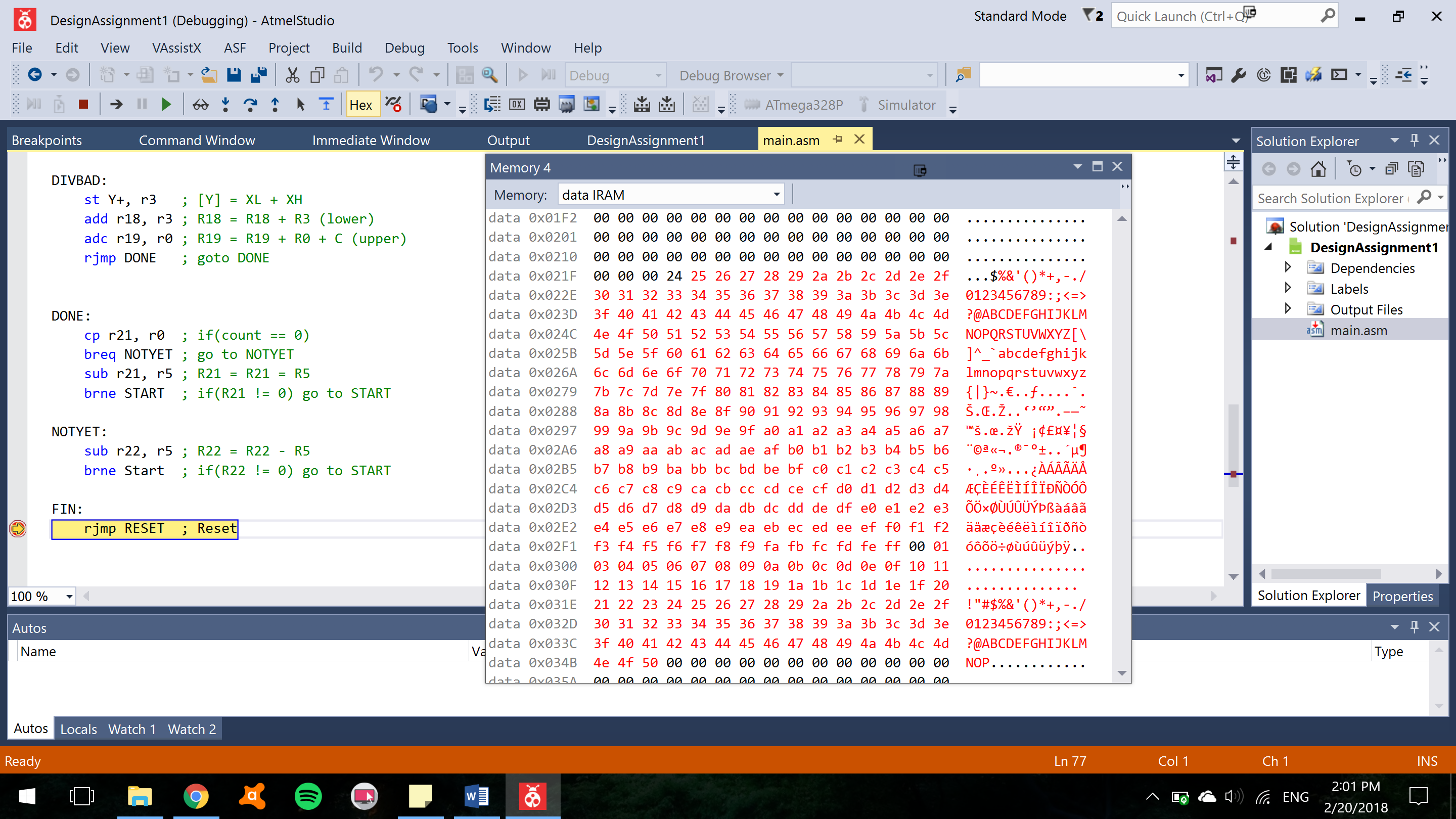


Figure Task one after storing values into addresses starting at 0x0222

TASK 2

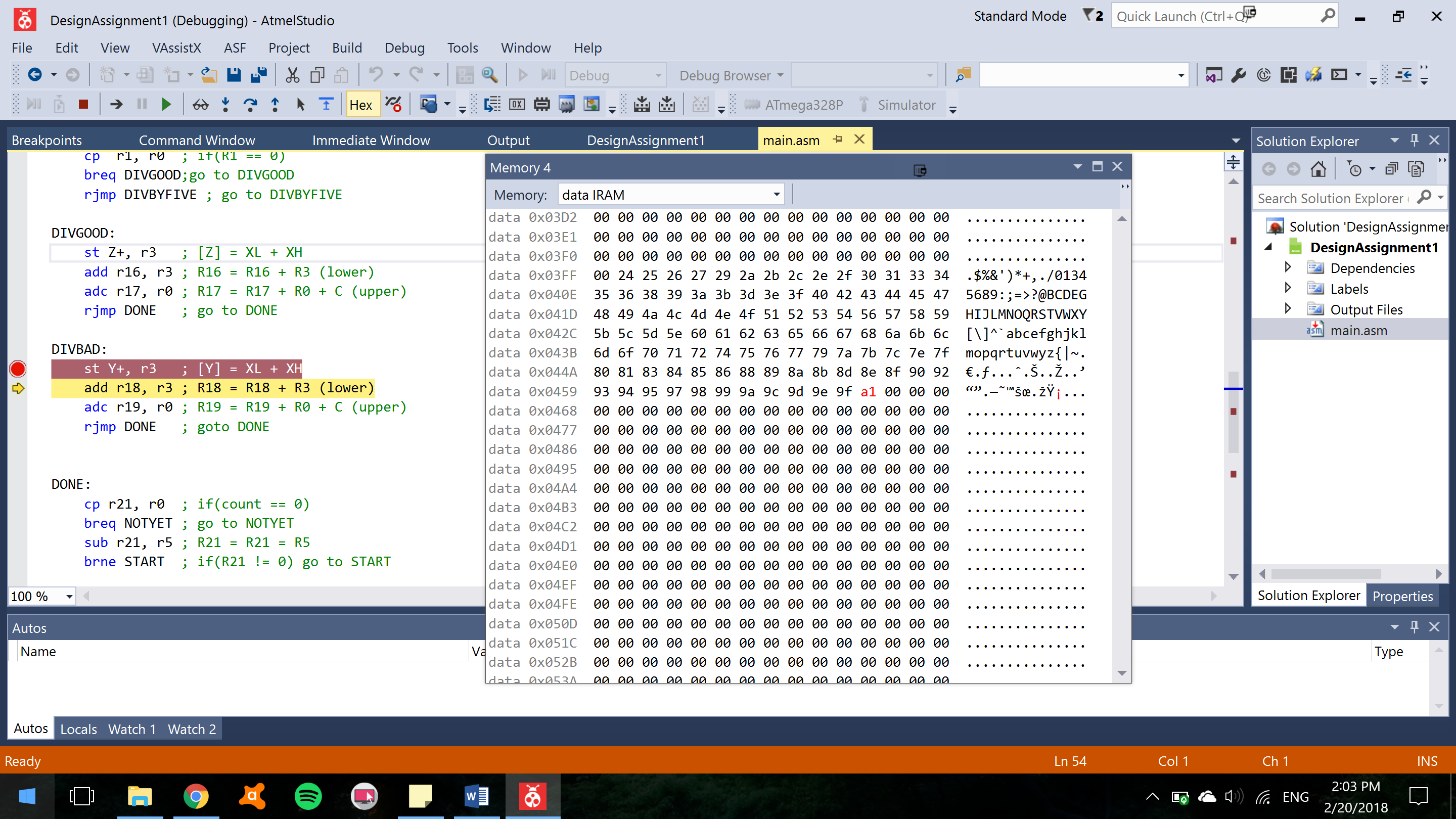


Figure Showing Y pointer while storing numbers

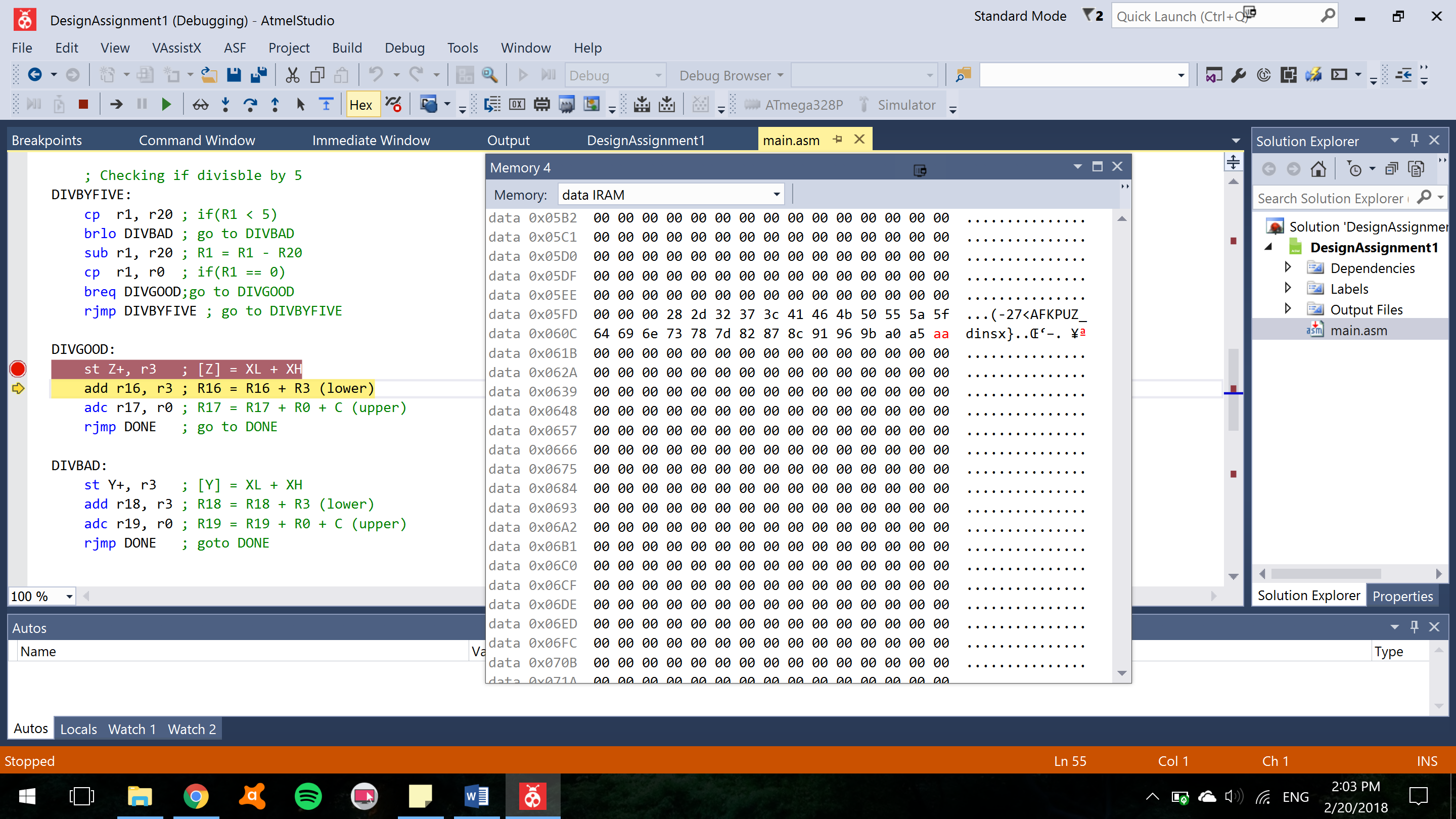


Figure Showing Z pointer while storing numbers

**Task 3**

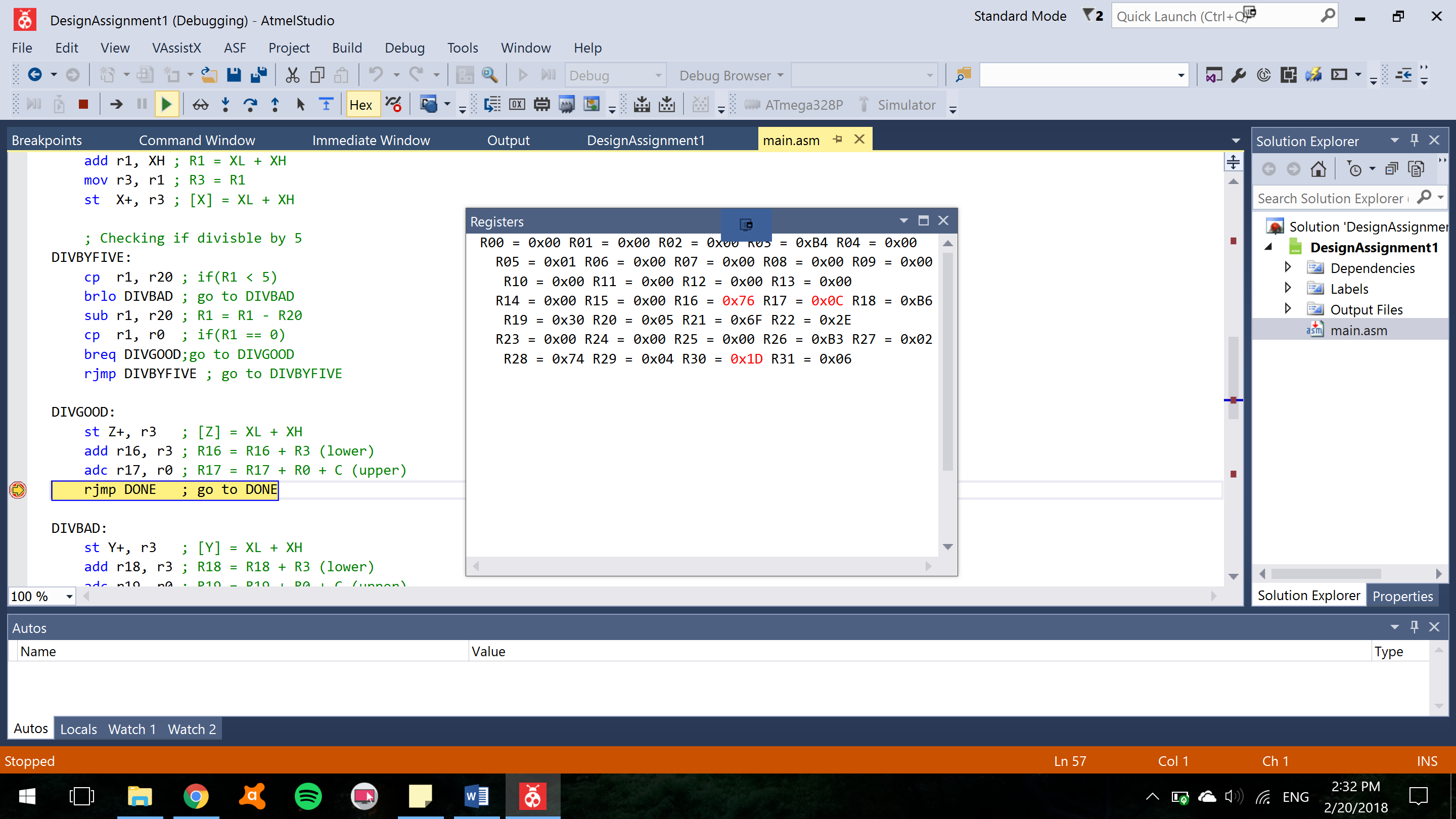


Figure R16:R17 after being added and stored

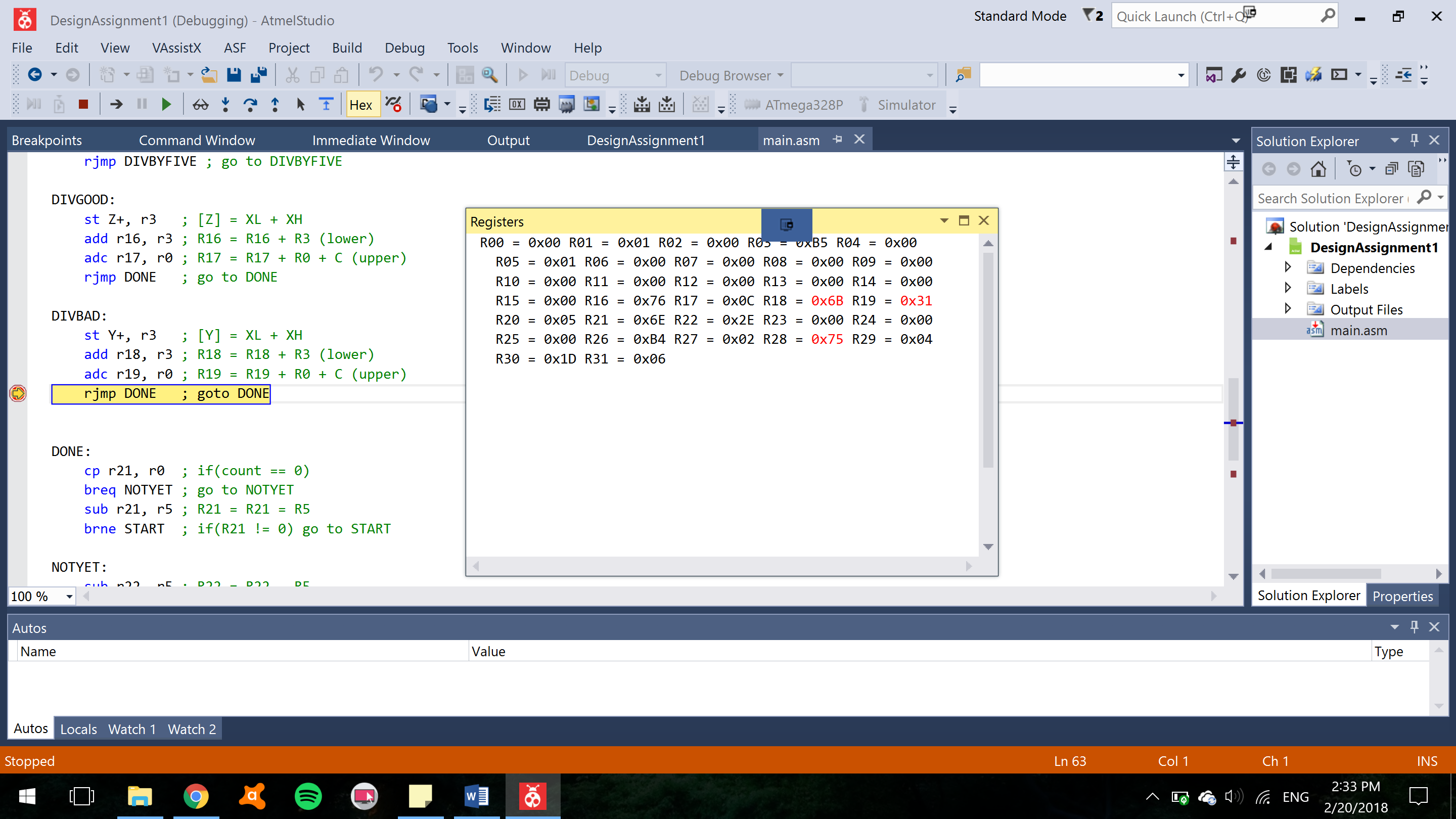


Figure R18:R19 after being added and stored

**TASK 4**

**TASK 5**

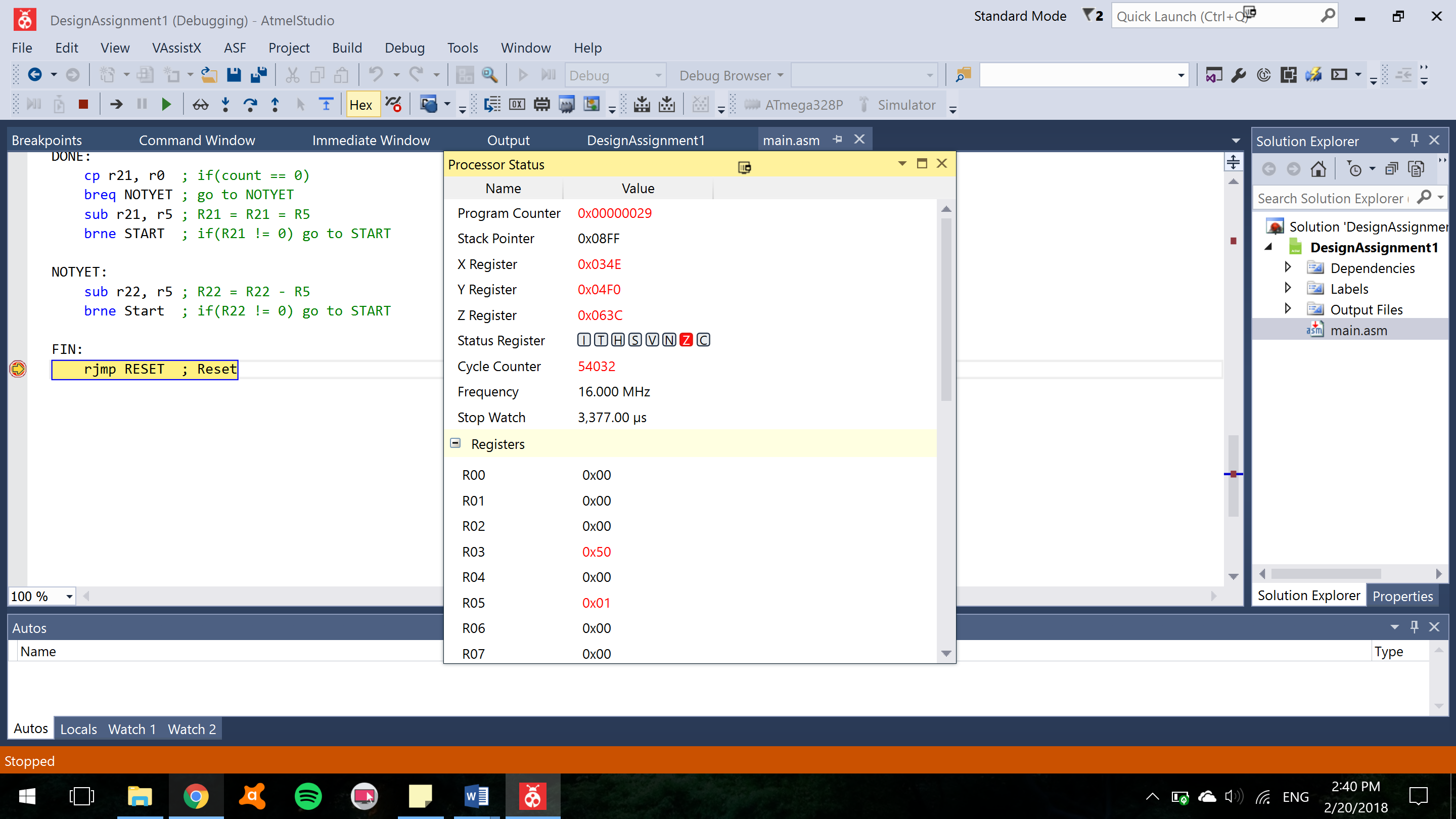
****

Figure Execution time of 3,360 us at a clock speed of 16 MHz. a Cycle counter of 54,032.

1. **SCREENSHOT OF EACH DEMO (BOARD SETUP)**

None.

1. **VIDEO LINKS OF EACH DEMO**

None

1. **GITHUB LINK OF THIS DA**

**Student Academic Misconduct Policy**

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

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

Trace Stewart