CPE301 – SPRING 2021

Design Assignment 1A

Student Name: Brian Wolak
Student #: 2000509437
Student Email: wolak@unlv.nevada.edu
Primary Github address: https://github.com/brianwolak/submission_da.git
Directory: DA_1A
Video Link: https://youtu.be/9ZRAsEbdSKc

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

Task 1:

Write a new program to perform a division of a 16-bit number with an 8-bit divisor. Use iterative subtraction to perform the above multiplication. Registers R17:R16 hold the 16-bit number, R18 hold 8-bit divisor, and R20- R19 holds the quotient, and R21 holds the reminder. Verify your result using a C program equivalent to your assembly code. Determine the execution time @ 16MHz/#cycles of your algorithm using the simulation.

Assembly Code:

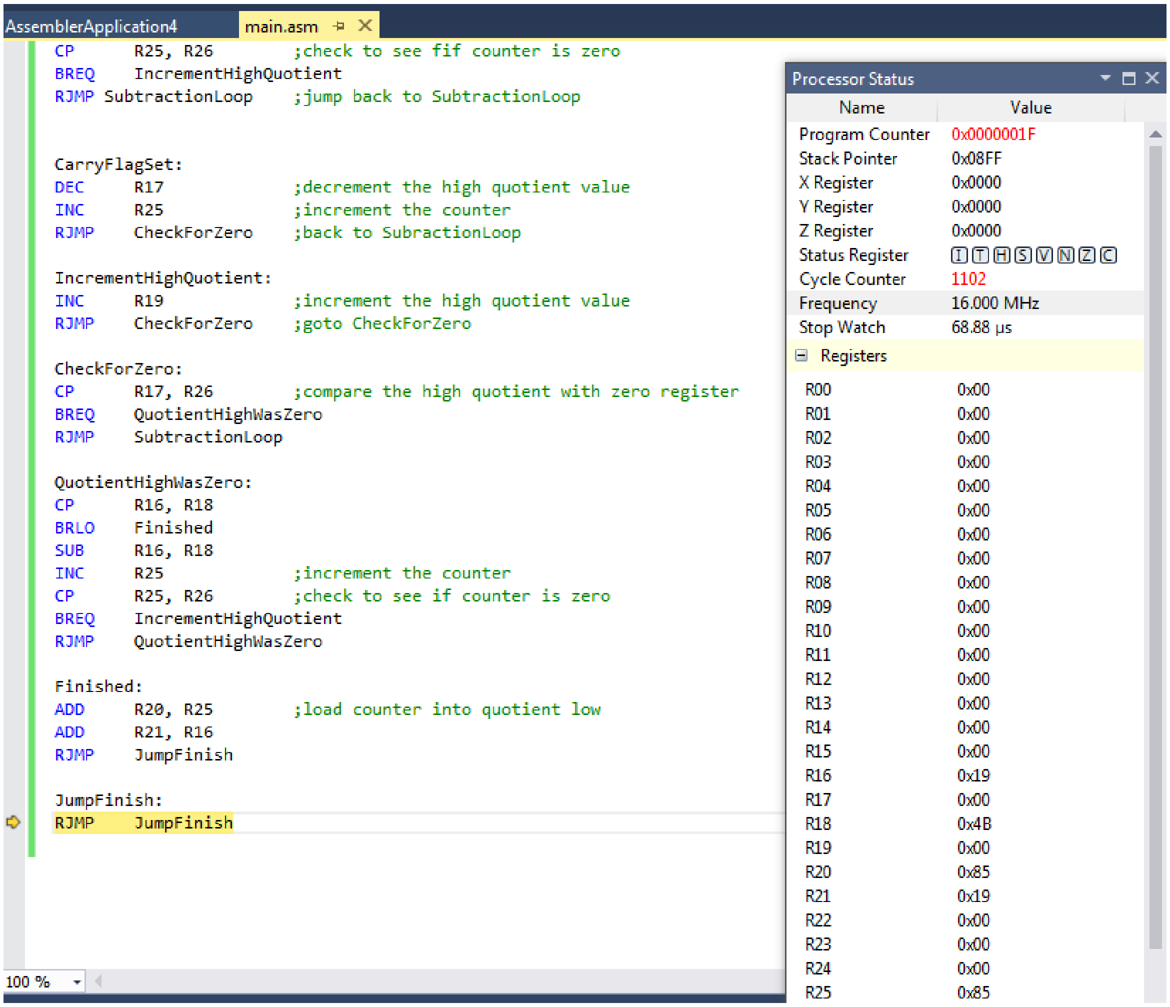
```
;Divisor value = 10000, or 0x2710 in hex
;Dividend value = 75, or 0x4B in hex
;Remainder value should equal 25 or 0x19 in hex

LDI        R17, 0x27        ;load divisor high of 27 into r17
LDI        R16, 0x10        ;load divisor low of 10 into r16
LDI        R18, 0x4B        ;load dividend of 0x4B into r18
LDI        R19, 0x00        ;load zero to result high
LDI        R20, 0x00        ;load zero to result low
LDI        R25, 0x00        ;counter register, load 0 into r25
LDI        R26, 0x00        ;storing a zero value

SubtractionLoop:
SUB        R16, R18        ;lower dividend subtracted with divisor
BRCS    CarryFlagSet       ;check to see if the carry flag is set
INC        R25             ;increment the counter
CP        R25, R26         ;check to see if counter is zero
BREQ    IncrementHighQuotient
RJMP SubtractionLoop       ;jump back to SubtractionLoop


CarryFlagSet:
DEC        R17              ;decrement the high divisor value
INC        R25              ;increment the counter
RJMP    CheckForZero       ;back to SubractionLoop

IncrementHighQuotient:
INC        R19              ;increment the high quotient value
RJMP    CheckForZero       ;goto CheckForZero
```

```
CheckForZero:
CP        R17, R26                 ;compare the high quotient with zero register
BREQ    QuotientHighWasZero
RJMP    SubtractionLoop

QuotientHighWasZero:
CP        R16, R18
BRLO    Finished
SUB        R16, R18
INC        R25                     ;increment the counter
CP        R25, R26                 ;check to see if counter is zero
BREQ    IncrementHighQuotient
RJMP    QuotientHighWasZero

Finished:
ADD        R20, R25                ;load counter into final quotient
ADD        R21, R16                ;load remainder
RJMP    JumpFinish

JumpFinish:
RJMP    JumpFinish
```

C++ Code:

```
#include <iostream>
#include <cmath>

using namespace std;

int main(){
        int quotient = 0;
        int divisor = 10000;
        int dividend = 75;
        int remainder = 0;

        quotient = divisor / dividend;
        remainder = divisor % dividend;

        cout << "10,000 / 75 = " ;
        cout << quotient ;
        cout << " with a remainder of ";
        cout << remainder << endl;

        return 0;
}
```

2. SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)

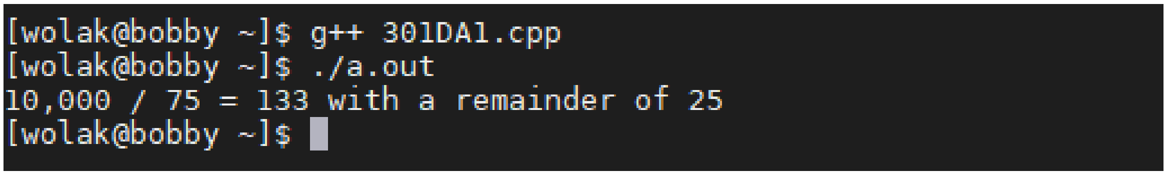


```
CP      R25, R26        ;check to see fif counter is zero
BREQ    IncrementHighQuotient
RJMP SubtractionLoop    ;jump back to SubtractionLoop


CarryFlagSet:
DEC     R17             ;decrement the high quotient value
INC     R25             ;increment the counter
RJMP    CheckForZero    ;back to SubractionLoop

IncrementHighQuotient:
INC     R19             ;increment the high quotient value
RJMP    CheckForZero    ;goto CheckForZero

CheckForZero:
CP      R17, R26        ;compare the high quotient with zero register
BREQ    QuotientHighWasZero
RJMP    SubtractionLoop

QuotientHighWasZero:
CP      R16, R18
BRLO    Finished
SUB     R16, R18
INC     R25             ;increment the counter
CP      R25, R26        ;check to see if counter is zero
BREQ    IncrementHighQuotient
RJMP    QuotientHighWasZero

Finished:
ADD     R20, R25        ;load counter into quotient low
ADD     R21, R16
RJMP    JumpFinish

JumpFinish:
RJMP    JumpFinish
```

Processor Status

Name	Value
Program Counter	0x0000001F
Stack Pointer	0x08FF
X Register	0x0000
Y Register	0x0000
Z Register	0x0000
Status Register	I T H S V N Z C
Cycle Counter	1102
Frequency	16.000 MHz
Stop Watch	68.88 µs
⊟ Registers	
R00	0x00
R01	0x00
R02	0x00
R03	0x00
R04	0x00
R05	0x00
R06	0x00
R07	0x00
R08	0x00
R09	0x00
R10	0x00
R11	0x00
R12	0x00
R13	0x00
R14	0x00
R15	0x00
R16	0x19
R17	0x00
R18	0x4B
R19	0x00
R20	0x85
R21	0x19
R22	0x00
R23	0x00
R24	0x00
R25	0x85

Atmel Studio output showing expected result & 68.88µs execution time



```
[wolak@bobby ~]$ g++ 301DA1.cpp
[wolak@bobby ~]$ ./a.out
10,000 / 75 = 133 with a remainder of 25
[wolak@bobby ~]$ 
```

C Code Confirmation of Results

3. Github Link to This Assignment

https://github.com/brianwolak/submission_da/tree/main/DA_1A