CPE301 – SPRING 2021

Design Assignment 1A

Student Name: Brian Wolak

Student #: 2000509437

Student Email: <wolak@unlv.nevada.edu>

Primary Github address:

Directory:

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;

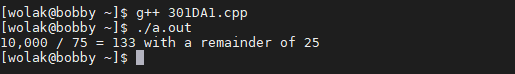
}

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

Graphical user interface, text

Description automatically generated

***Atmel Studio Output Showing Expected Result***



***C Code Confirmation of Results***

1. **GITHUB LINK OF THIS DA**