A simple program

With the mon and add instrations we can we can wite simple programs.
The following is a simple program which adds Two numbers.

Disassembly of section .text:

00000000	< ma	ain:	>:		
0:	0 6	00	fa	lnk	#0x6
2:	50	f8	2f	mov.W	#0xff85, w0
4:	00	0f	78	mov.w	w0, [w14]
6 :	e0	02	20	mov.w	#0x2e, $w0$
8:	10	07	98	mov.w	w0, [w14+2]
a:	1e	00	90	mov.w	[w14+2], $w0$
c:	1e	00	40	add.w	w0, [w14], w0
e:	20	07	98	mov.W	w0, [w14+4]
10:	00	00	eb	clr.w	w0
12:	00	80	fa	ulnk	
14:	00	00	06	return	

How each instruction is on a line that begins with a nex number of the words the instruction is offset from the first instruction, then the 24-bit instruction withen in hex, then the assembly innemonic, and lastly any operands. For example, the 3.7 line

relative address 2M-bit __ mnemonic opercands 31

Here the Ink and wink commands can be ignowed, they relate to the structure of the in C program we used to generale This code. The return instruction should be clear, it returns program execution to wherever it orginated, which we will discuss later.

Linex 2 and 3 contain Two Move instructions, hine 2 copies the 16-bit literal into register wo (In fact this literal represents a negative integer in 20 representation, -123 decimal)

Line 3 then moves this to an addrew given by the Nalve in W14.

If you reference the programmers
you will see WIH is used to store the
frame pointer, the frame pointer is the
starting endoress for the region in data
memory allocated to this program.

The next two lines do a similar operation, storing 0x2e (46 decimal in the following word of delice memory

	q		
After these 4 ins	struction	ès we	, have
the following			addresses
	0x85	w14	frame
WO 0x002e	Oxff	WIMTI	
	oxle	W14+2	/
frame pointer is	0x00	W14+3	
in wi4	?	W14+4	
	2	w14+4	
		1	
The and instruction	i noeu	exect	tes and
The add instruction adds the Two 16	-bit op	UZOM =	1s
add i	wo, [w	14 J. u	00
4		ニノ ・	√ .
regular or and signer and experience of the contract of the co	Scoultice	Ole	strinal tor

source destination registers registers

(IWMJ) + 0x002e 46 WO 0xffb3 -77

This instruction overwrites the contents of wo with the result, and then moves it to the address will + 4, and then was the clear word instructor to clear (set to 0x0000) wo.

33

es annua per menera e s en res que anteque per competible en may magnica an en existención de en el entre en e	Upon completion of these instructions
Marmanturry and retry or year and to have been provided and the control of the co	we have
man giringingan pangganggan giga yang pandah pinyakakan manala karabi sinta kita asta asta asta asta asta asta	
	0 S 0x85 WIH
nemente van delt e stad het de	0x0000 0x55 WIMTI
and a photographic control of a photographic control of the photographic control of th	h S 0x2e w14+2
and the second section of the second section of the second section of the second section secti	0x00 W14+3
one of the second secon	0 S 0x 63 WIM+ 4
	(0x5f W14+5
naka manamini kalamanan jaman kang (pyana) di diri di di di di di diri di	
antaus d'authrité de l'implementation de la place agrésse de la conférence de l'implementation de l'implementa	
	As an axide Hús assembly was generated by the XC16 C compiler for the C code
	agnerated by the XCI6 C compiler
ar vinner den 13 meneror egit eri eri egen, gond de milje in blet i meneror de tribene e de en en en en en en	Por the Code
er voormalier voor deep vanderde jaar voormeen deer Bevoorde Pier aan militer een voormeen voormeen voormeen v	
Bending bendings to the second	
et var en de artistrio Unività a unitano eggi vi anno undebre estrebbera et meser	int main(void) {
mendabilan vilmustrusi (musika periode militaria) (militaria) (militaria) (militaria) (militaria) (militaria)	int a,b,c;
ердін, шатана унта у да у даную турка у ушарання шада ақынаўся сал төннөмей этой ойній Але	
на част та серина ти серина по сето серина за пред ти серина по да се	a = -123; b = 46;
n aller allen engere ig terminente en kombe (spiler), som kombe (spiler), som en ste (spiler), ste (spiler), s	c = a + b;
e de la companya de La companya de la companya de	
· (***********************************	return 0;
t yang garandari katan di di katan di dikatan di dikatan di	· · · · · · · · · · · · · · · · · · ·
h midrigilde (() die zu Musjela under of de didpost of unit (e-1160 y Ministe Me) forener	

>xc16-objdump -d newmainXC16.o

Disassembly of section .text:

```
00000000 < main>:
   0:
          06 00 fa
                                     #0x6
                          lnk
   2:
          50 f8 2f
                                     #0xff85, w0
                          mov.w
   4:
          00 Of 78
                                     w0, [w14]
                          mov.w
   6:
          e0 02 20
                          mov.w
                                     #0x2e, w0
                                     w0, [w14+2]
   8:
          10 07 98
                          mov.w
                                     [w14+2], w0
   a:
          1e 00 90
                          mov.w
                                     w0, [w14], w0
   c:
          1e 00 40
                          add.w
          20 07 98
   e:
                          w.vom
                                     w0, [w14+4]
  10:
          00 00 eb
                          clr.w
                                     w0
  12:
          00 80 fa
                          ulnk
  14:
          00 00 06
                          return
```

```
int main(void) {
    int a,b,c;
    a = -123;
    b = 46;
    c = a + b;
    return 0;
}
```

We can for their investigate this program.

There are 11 instructions, each instruction is 3 bytes in size, so we expect this program to require 33 bytes or 2/3×33 words of program memory

For the PICRYFJ64GA002 program flash memory is located from 0x000200 to 0x00,00 fe. The XC16 compiler will place this code at 0x00020C Castitional code generated by the XC16 C compiler occupies the program memory below this, there is 252 bytes for this code)

The PIC24FJ64G-A002 has Date RAM
from 0x0800 to , The XC16 compiler
will put NIH = 0x0806, place the Nature
of a at this address, b at 0x0808, and
c at 0x80a, These are book variables
for this program.

How long does it take this program code to execute on a PIC24FJ64GA002 with a 16 MHz instruction clock frequency?

From the Programmer's Reforence Manual we know that each instruction in this code, except RETURN, takes I instruction cycle to execute. The RETURN instruction normally takes 3 instruction cycles.

With the 16 NHz instruction clock frequency, Feyc=16 MHz, the instruction clock period is Teyc= /16MHz=62.5 ns.

So for the program execution Time

(10×1+1×3).62.5 ns = 812,5 ns