บทที่ 4

การสร้าง, การทดสอบและผลการทดสอบการทำงานของระบบประมวลผล

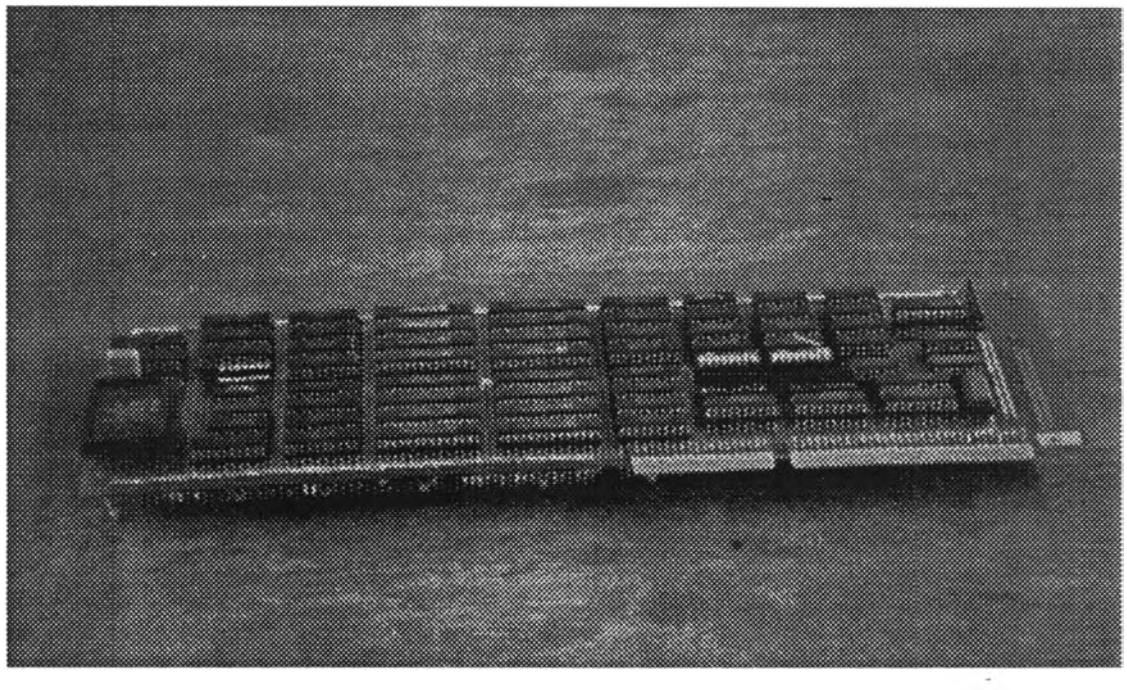
ในบทที่ 3 ได้กล่าวถึงการออกแบบระบบซึ่งรวมทั้งการออกแบบวิธีการทดสอบการทำงานของส่วนต่างๆของระบบประมวลผลทั้งทางด้านฮาร์ดแวร์และซอฟต์แวร์ ในบทนี้จะกล่าวถึงการสร้าง, การทดสอบและผลการทดสอบการทำงานของระบบประมวลผลที่ออกแบบและสร้างขึ้น

4.1 การสร้างระบบประมวลผล

ในการสร้างระบบประมวลผลสัญญาณสามารถแบบออกได้เป็น 2 ส่วนดังนี้

4.1.1 การสร้างระบบประมวลผลทางด้านฮาร์ดแวร์

เนื่องจากจำนวนสายไฟและอุปกรณ์ที่ใช้มีจำนวนมากทำให้การสร้างฮาร์ดแวร์ได้สร้างและประกอบฮาร์ดแวร์ลงบนแผ่นวงจรพิมพ์ (PCB) ไม่สามารถทำได้ดังนั้นจึงได้สร้างบอร์ดและทำการเดินสายแบบ Wired Wrap ดังแสดงในรูปที่ 4.1 และรายการของอุปกรณ์ทั้งหมดที่ใช้ในแสดงได้ดังตารางที่ 4.1



รูปที่ 4.1 ฮาร์ดแวร์ของระบบประมวลผลที่ออกแบบและสร้างขึ้น

ตารางที่ 4.1 รายการอุปกรณ์ทั้งหมดที่ใช้รวมทั้งจำนวนของอุปกรณ์

เบอร์ของ IC ที่ใช้ในการสร้างระบบประมวลผล	จำนวน (ตัว)
TMS320C25	1
MT5C2568	16
ADC0804	1
74LS00	2
74LS04	2
74LS28	2
74LS32	2
74LS74	1
74LS90	2
74LS93	1
74LS125	1
74LS138	2
774LS175	2
74LS245	18
74LS257	2
74LS374	6
74LS688	2
LF398	1
MF6	1
DIP SWITCH-8	2
CRYSTAL 20 MHz	1
R PACK 9 (220K)	3
R (1K)	1
R (10K)	1
R (100K)	1
C (0.1 μf)	60
C (100pf)	1
C (10pf)	2
C (1nf)	1

4.1.2 การสร้างระบบประมวลผลในส่วนของด้านซอฟต์แวร์

ซอฟต์แวร์ในส่วนที่ใช้ในการควบคุมระบบประมวลผลจะแบ่งออกเป็น 2 ส่วนดังนี้

ซอฟต์แวร์ในส่วนของภาษาซีของบริษัทไมโครซอฟต์โดยในการพัฒนาโปรแกรมจะทำการพัฒนาเป็นฟังก์ชันโดยจะมีฟังก์ชันที่ใช้สำหรับการทำงานต่างๆ เพื่อความสะดวกและความยืดหยุ่นในการพัฒนาโปรแกรมและในการนำมาร่วมกันเป็นโปรแกรมจะเรียกใช้เฉพาะฟังก์ชันที่จำเป็นต้องใช้ในโปรแกรมเนื่องจากข้อจำกัดของการทำงานบนระบบปฏิบัติการ MS-DOS ดังนั้นจึงไม่สามารถสร้างโปรแกรมที่มีขนาดใหญ่มากได้

ซอฟต์แวร์ในส่วนของภาษาแอสเซมบลีสำหรับ TMS320C25 ของบริษัท TI โดยในการพัฒนาโปรแกรมจะทำในลักษณะเดียวกับในส่วนของภาษาซี

4.2 การทดสอบและผลการทดสอบการทำงานของระบบประมวลผล

การทดสอบการทำงานของระบบประมวลผลแบ่งได้เป็น 2 ระดับคือการทดสอบการทำงานระดับล่าง (Low Level) และการทดสอบการทำงานระดับบน (High Level) ดังนี้

4.2.1 การทดสอบและผลการทดสอบการทำงานของระบบประมวลผลระดับล่าง

ในการทดสอบระดับนี้จะเป็นการทดสอบการทำงานขั้นพื้นฐานทางด้านฮาร์ดแวร์ของระบบประมวลผล หลังจากที่ได้ทำการสร้างบอร์ดแล้วได้ทำการตรวจสอบความถูกต้องของสายไฟของวงจรด้วยโอห์มมิเตอร์ก่อนที่จะติดตั้งอุปกรณ์ลงไป หลังจากนั้นจึงทำการทดสอบฮาร์ดแวร์ที่สร้างขึ้นด้วยวิธีที่ได้ออกแบบไว้และแก้ไขจนทำงานได้ถูกต้อง

เมื่อทำการตรวจสอบการทำงานขั้นพื้นฐานจนระบบสามารถทำงานได้ถูกต้องซึ่งจะต้องทำก่อนที่จะสร้างซอฟต์แวร์ที่ใช้ทดสอบการทำงานของระบบประมวลผลขั้นพื้นฐาน เนื่องจากการสร้างและทดสอบฮาร์ดแวร์ไม่สมบูรณ์จะทำให้ไม่สามารถสร้างฟังก์ชันในระดับที่ทำงานเกี่ยวข้องกับฮาร์ดแวร์ได้ ซึ่งจะส่งผลต่อไปถึงการสร้างฟังก์ชันในระดับที่สูงขึ้น เมื่อทำการทดสอบการทำงานขั้นพื้นฐานของระบบประมวลผลด้วยโปรแกรมที่ได้สร้างขึ้นมาจนระบบประมวลผลสามารถทำงานได้ถูกต้อง จึงจะทำการทดสอบการทำงานระดับสูงต่อไป

กล่าวโดยสรุปสำหรับการทดสอบในส่วนนี้ระบบประมวลผลสามารถทำงานได้ตามที่ได้ออกแบบไว้ถึงแม้ว่าจะมีปัญหาเกี่ยวกับสัญญาณรบกวนที่ทำให้ระบบขาดความเสถียรในการทำงานไปบ้าง

4.2.2 การทดสอบการทำงานของระบบประมวลผลระดับสูง

หลังจากที่ได้สร้างและทดสอบฮาร์ดแวร์ของระบบประมวลผลระดับล่างที่ออกแบบขึ้น ได้ทำการแก้ไขและปรับแต่งจนระบบประมวลผลสามารถทำการรับส่งข้อมูลกับเครื่อง PC, แปลงสัญญาณแอนะลอก

เป็นสัญญาณดิจิตอลและรวมไปถึงการประมวลผลในภาคแอนะลอกโดยไม่มีความผิดพลาดได้ตามความต้องการแล้วจึงจะทำการทดสอบในระดับสูงต่อไป โดยสามารถแบ่งการทดสอบออกเป็นเป็น 3 ขั้นตอนดังนี้

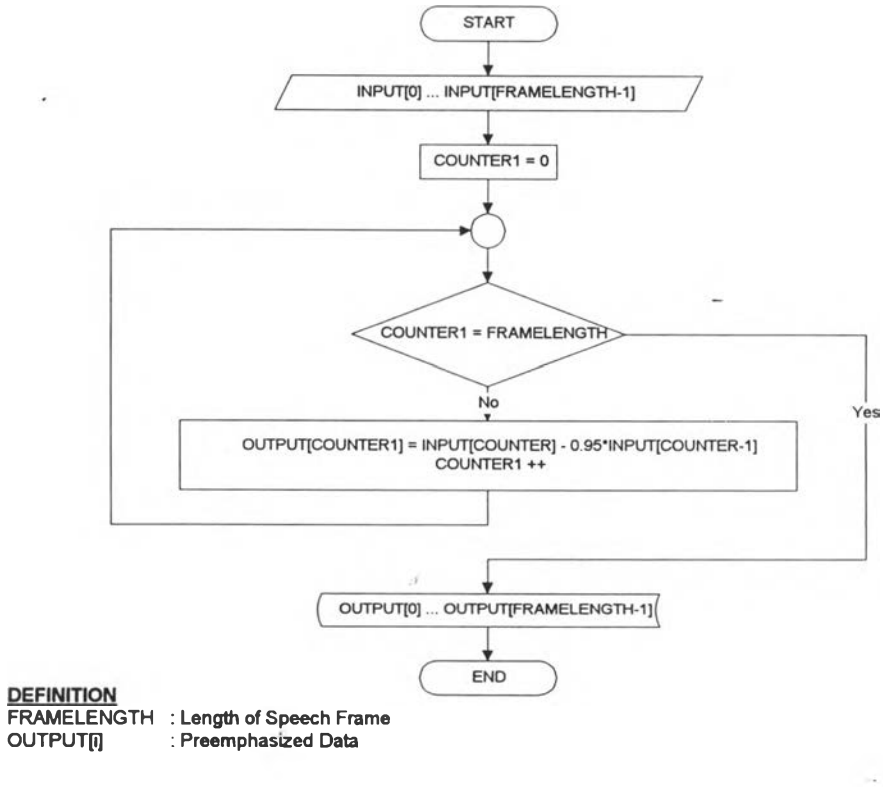
4.2.2.1 การกำหนดโปรแกรมตัวอย่าง

ในขั้นตอนนี้เราใช้ตัวอย่างโปรแกรมการทำ Preemphasized ซึ่งเป็นวงจรกรองแบบดิจิตอลแบบ first order [8] เพราะมีอัลกอริทึมซึ่งมีความซับซ้อนไม่มากดังแสดงในสมการที่ 4.1

$$H(z) = 1 - aZ^{-1} \tag{4.1}$$

โดย a = 0.95

ในการทดสอบจะใช้ผลการคำนวณที่ได้จากระบบซึ่งได้ทำการออกแบบเปรียบเทียบกับการคำนวณซึ่งได้จากเครื่อง PC โดย Flowchart ของโปรแกรมนี้สามารถแสดงได้ดังในรูปที่ 4.2 และตัวอย่างโปรแกรมซึ่งเป็นภาษา Visual C++ (สำหรับเครื่อง PC) และภาษาแอสเซมบลีของ TMS320C25 (สำหรับระบบ) แสดงในรูปที่ 4.3 และรูปที่ 4.4



รูปที่ 4.2 Flowchart ของตัวอย่างโปรแกรม Preemphasized ซึ่งใช้ในการทดสอบระบบ

```
void Pre_Emphasize_Menu( void );
void Pre_Emphasize_Menu( void )
{
char            Input_FileName[_MAX_PATH], Raw_Output_FileName[_MAX_PATH],
                FileName[_MAX_FNAME], Drive[_MAX_DRIVE],
                Directory[_MAX_DIR], Extension[_MAX_EXT];
int             Data, X0, X1;
long            FileLength, Wave_Header_Length, i;
unsigned char   Select;
FILE            *Input_FilePtr,  *Raw_Output_FilePtr;
printf( "Enter Input File Name\t\t: " );
scanf( "%s", Input_FileName );
_splitpath( Input_FileName, Drive, Directory, FileName, Extension );
_makepath( Raw_Output_FileName, Drive, Directory, FileName, ".raw" );
Select = 'z';                           /* Initialize */
Wave_Header_Length          = 22;       /*  wave file have header length = 44
byte */
while( Select != '0' )
{               _clearscreen( _GCLEARSCREEN );
                _printf("Pre_Emphasize Menu\n" );
                _printf( "-----------------------\n" );
                printf( "1. Change Input File Name\n" );
                printf( "2. Change Raw Output File Name\n" );
                printf( "3. Pre Emphasize Process (Int)\n" );
                printf( "0. Exit\n\n" );
                printf( "Input File Name\t\t\t: | %s |\n", Input_FileName );
                printf( "Raw Output File Name\t\t: | %s |\n", Raw_Output_FileName );
                FileLength  = FileLength_Word( Input_FileName );
                printf( "Length of Input File\t\t: | %u Word |\n", (unsigned int)(FileLength)
);
                Select = _getch();
switch( Select )
{
case '1'
{               printf( "Enter Input File Name\t\t: " );
                scanf( "%s", Input_FileName );
                FileLength  = FileLength_Word( Input_FileName );
} break;
case '2'
{               printf( "Enter Raw Output File Name\t\t: " );
                scanf( "%s", Raw_Output_FileName );
} break;
case '3'
{               Input_FilePtr     = fopen( Input_FileName, "rb" );
                Raw_Output_FilePtr = fopen( Raw_Output_FileName, "wb" );
FileLength      = FileLength_Word( Input_FileName ) - (long)( Wave_Header_Length );
                fseek( Input_FilePtr, 0, SEEK_SET );
                fseek( Raw_Output_FilePtr, 0, SEEK_SET );
                for( i = 0; i < (long)(Wave_Header_Length); i++ )
                        _getw( Input_FilePtr );
                X0 = 0;
                for( i = 0; i < FileLength; i++ )
                {               X1              = _getw( Input_FilePtr );
                                Data            = (int)( X1 - (0.95)*X0 );
                                _putw( Data, Raw_Output_FilePtr );
                                X0              = X1;
                }
                fclose( Input_FilePtr );
                fclose( Raw_Output_FilePtr );
                printf( "\nCalculation Complete");
                getch();
} break;
}}}
void main()
{               Pre_Emphasize_Menu();
}
```

รูปที่ 4.3 ตัวอย่างโปรแกรม Preemphasized ซึ่งใช้ในการทดสอบระบบ (ภาษา Visual C++)

```
title   "PREEMPHASIZED BY DIGITAL FILTER"
text
                                ; INITIALIZATION
LDPK    0                       ; Set Data Page Pointer to PAGE0 ;
LALK    a                       ;                               ;
SACL    COEFF                   ; Set the Cofficient to 0.95    ;
LRLK    AR0, OLD_DATA           ; Load address of s(-1)         ;
LRLK    AR1, PAGE10             ; Load address of s(n-1)        ;
LRLK    AR2, PAGE10             ; Load address of s(n)          ;
LRLK    AR3, PAGE12             ; Load Address of preemphasized data ;
                                ; PROCESS 1
                                ;CALCULATE : s(0) = s(0) - ( a * s(-1) );
LARP    AR0                     ;                               ;
LT      *                       ; Load T Register with x(-1)    ;
MPY     COEFF                   ;P Register = ( a * s(-1) ) in Q30 Format;
PAC                             ;                               ;
SACH    TEMP1, 1                ; TEMP1 = ( a * s(-1) ). in Q15 Format ;
LARP    AR2                     ;                               ;
LAC     *+                      ; Load Accumulator with x(0)    ;
SUB     TEMP1                   ;                               ;
ROR                             ;                               ;
LARP    AR3                     ;                               ;
SACL    *+                      ; Save s(0) in PAGE12. in Q14 Format ;
LACK    FRAMELENGTH_1           ;                               ;
SACL    COUNTER1                ;Set the COUNTER to FRAMELENGTH-1;
FILTER  LARP    AR0             ; PROCESS 2
                                ; CALCULATE : s(n) = s(n)-(a*s(n-1)) ;
                                ; WHEN n = 1 to FRAMELENGTH-1 ;
LT      *+                      ; Load T Register with x(n-1)   ;
MPY     COEFF                   ; P Register=(a*x(n-1)) in Q30 Format ;
PAC                             ;                               ;
SACH    TEMP1, 1                ; TEMP1=(a*s(n-1))in Q15 Format ;
LARP    AR1                     ;                               ;
LAC     *+                      ; Load Accumulator with s(n)    ;
SUB     TEMP1                   ;                               ;
ROR                             ;                               ;
LARP    AR2                     ;                               ;
SACL    *+                      ;Stored s(n) in PAGE10 in Q14 Format ;
LAC     COUNTER1                ;                               ;
SUBK    ONE                     ;                               ;
SACL    COUNTER1                ; Decrement COUNTER1            ;
BNZ     FILTER                  ; If COUNTER is not equal to ZERO ;
                                ;               Goto FILTER     ;
                                ; Else                          ;
                                ;               End of the routine ;
                                ;-------------------------------;
```

รูปที่ 4.4 ตัวอย่างโปรแกรม Preemphasized ซึ่งใช้ในการทดสอบระบบ (ภาษาแอสเซมบลี)

4.2.2.2 การกำหนดข้อมูลซึ่งจะใช้ในการประมวลผลโดยโปรแกรมตัวอย่าง

ข้อมูลซึ่งจะใช้ในการทดสอบระบบประมวลผลจะแบ่งได้เป็น 2 ส่วนคือ

- ข้อมูลตัวอย่างซึ่งได้จากการสัญญาณคลื่นไซน์ซึ่งมีขนาด Vpp = 0-5 V (จาก Function Generator)
- ข้อมูลตัวอย่างซึ่งได้จากสัญญาณคลื่นเสียงพูด (จาก File เสียงพูดซึ่งถูกบันทึกในห้องวิจัยปฏิบัติการกรรมวิธีสัญญาณดิจิตอล)

4.2.2.3 เปรียบเทียบผลการคำนวณระหว่างระบบประมวลผลกับโปรแกรมซึ่งทำการคำนวณบนเครื่อง PC

การแสดงผลการคำนวณสามารถแบ่งได้เป็น 2 ส่วนตามข้อมูลซึ่งนำมาทดสอบดังนี้

- ผลการคำนวณที่ได้จากข้อมูลตัวอย่างซึ่งเป็นสัญญาณคลื่นไซน์ที่ความถี่ต่างๆ

ผลการคำนวณจะสามารถแสดงได้ดังรูปที่ 4.5 และตารางที่ 4.1 โดยเป็นการเปรียบเทียบผลการคำนวณ Preemphasized ระหว่างระบบซึ่งได้ทำการออกแบบกับการคำนวณจากเครื่อง PC วิธีการเปรียบเทียบที่ใช้ในการวิจัยนี้เป็นการหาค่าความคลาดเคลื่อนในแบบ NMSE (Normalized Mean Square Error) ซึ่งผลการคำนวณที่ได้จากการทำงานของระบบที่ออกแบบเทียบกับผลการคำนวณด้วยหน่วยประมวลผลกลางของเครื่องไมโครคอมพิวเตอร์ โดยค่า NMSE สามารถหาได้จากสมการที่ (4.2)

$$NMSE = \left(\frac{\sum_{i=1}^{N} (x_i - y_i)^2}{\sum_{i=1}^{N} x_i^2} \right) \times 100 \qquad (4.2)$$

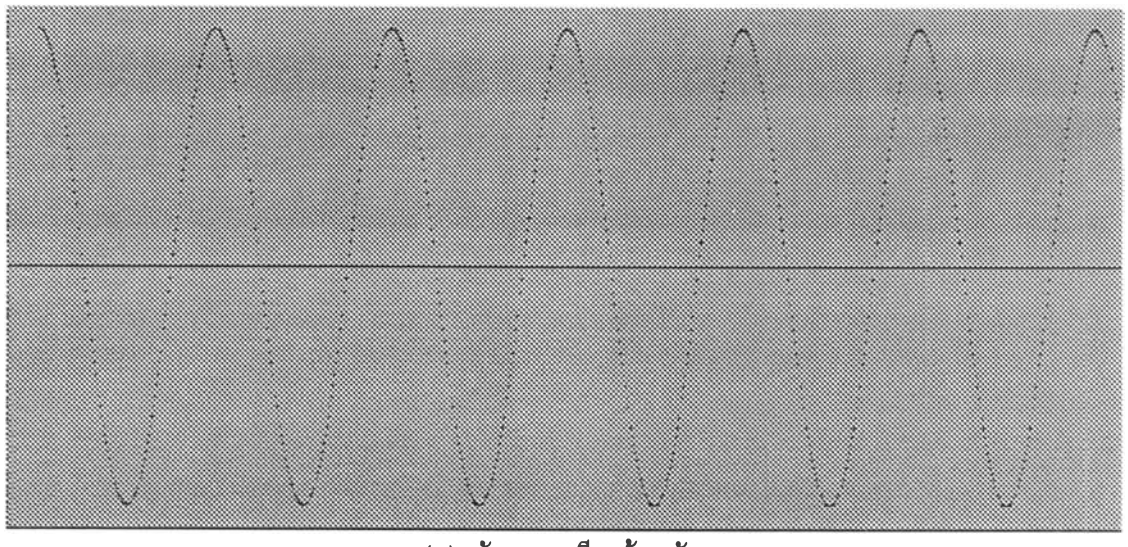
โดย

- $NMSE$ มีหน่วยเป็นเปอร์เซ็นต์
- x_i คือข้อมูลซึ่งผ่านการคำนวณโดยเครื่อง PC
- y_i คือข้อมูลซึ่งผ่านการคำนวณโดยระบบประมวลผล

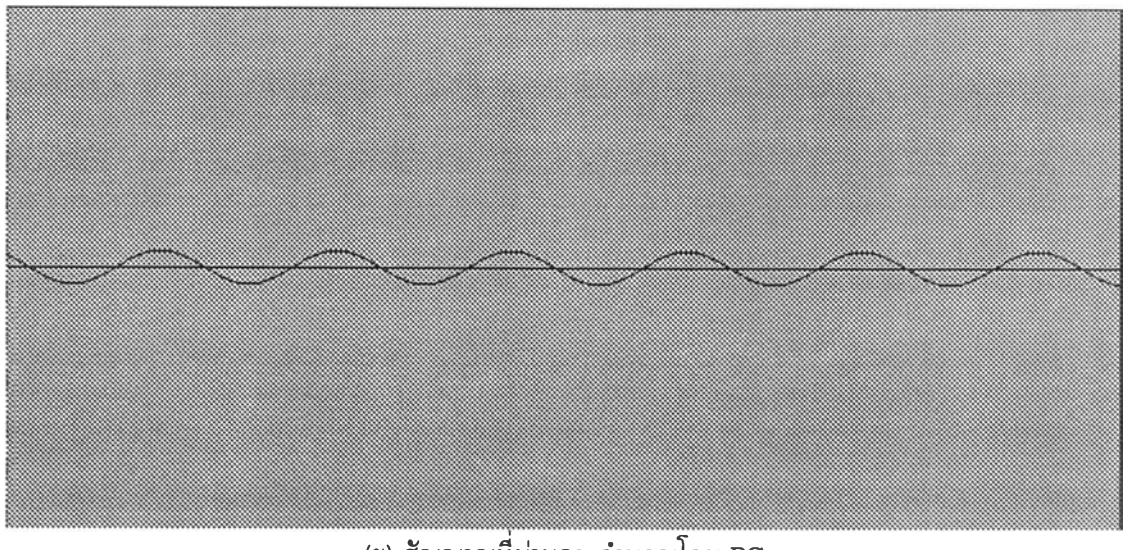
ตารางที่ 4.1 ค่า NMSE ของผลลัพธ์ที่ได้จากระบบกับผลลัพธ์ที่ได้จากเครื่อง PC (ข้อมูลตัวอย่างซึ่งได้จากสัญญาณคลื่น Sine จำนวน 5000 ตัวอย่าง)

ค่าความถี่ของสัญญาณ Sine (KHz)	NMSE ($\times 10^{-6}$ %)
0.5	0.361
1	0.334
3	0.325
5	0.354

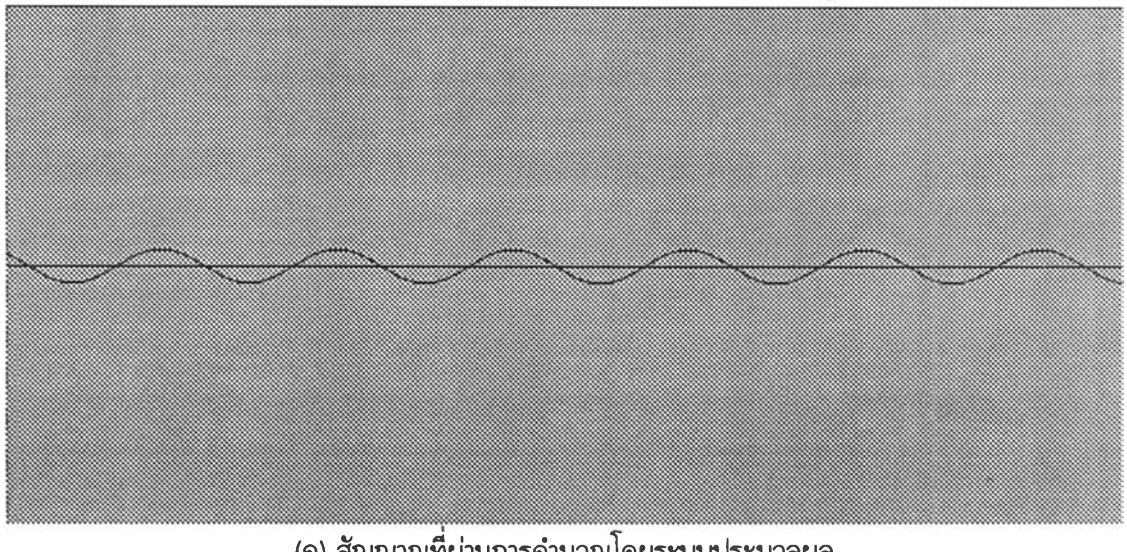
จากตารางจะเห็นได้ว่าผลต่างที่ได้จากการคำนวณทั้งสองวิธีมีค่า NMSE ไม่เท่ากับ 0 นั้นแสดงว่าในการคำนวณมีความคลาดเคลื่อนเกิดขึ้นจึงทำให้ค่าที่ได้จากการคำนวณทั้งสองวิธีได้ผลลัพธ์ที่แตกต่างกัน และค่า NMSE มีค่าอยู่ในช่วงเดียวกันและมีค่าไม่มากแสดงให้เห็นว่าความคลาดเคลื่อนในการคำนวณเกิดการความไม่ละเอียดของระบบประมวลผล



(ก) สัญญาณเสียงต้นฉบับ



(ข) สัญญาณที่ผ่านการคำนวณโดย PC



(ค) สัญญาณที่ผ่านการคำนวณโดยระบบประมวลผล

รูปที่ 4.5 เปรียบเทียบสัญญาณ Sine จากการคำนวณทั้ง 2 วิธีกับสัญญาณ Sine ต้นฉบับ

- ผลการคำนวณที่ได้จากข้อมูลตัวอย่างซึ่งได้จากการสัญญาณคลื่นเสียงพูด

ผลการคำนวณจะสามารถแสดงได้ดังรูปที่ 4.6 และตารางที่ 4.2 โดยเป็นการเปรียบเทียบการผลคำนวณ Preemphasized ระหว่างระบบซึ่งได้ทำการออกแบบกับการคำนวณจากเครื่อง PC โดยในการเปรียบเทียบเราใช้ค่า NMSE

ตารางที่ 4.2 ค่า NMSE ของผลลัพธ์ที่ได้จากการระบบกับผลลัพธ์ที่ได้จากเครื่อง PC
(ข้อมูลตัวอย่างซึ่งได้จากการสัญญาณคลื่นเสียงพูดคำว่า "หนึ่ง")

ลำดับที่ของ Frame ของสัญญาณเสียงพูด	ความยาวของ Frame (ms)	NMSE ($x10^{-3}$%)
1	20	0.162
	30	0.167
	40	0.163
5	20	0.004
	30	0.001
	40	0.001
10	20	0.001
	30	0.001
	40	0.007

ตารางที่ 4.3 ค่า NMSE ของผลลัพธ์ที่ได้จากการระบบกับผลลัพธ์ที่ได้จากเครื่อง PC
(ข้อมูลตัวอย่างซึ่งได้จากการสัญญาณคลื่นเสียงพูดคำว่า "สอง")

ลำดับที่ของ Frame ของสัญญาณเสียงพูด	ความยาวของ Frame (ms)	NMSE ($x10^{-3}$%)
1	20	0.284
	30	0.273
	40	0.274
5	20	0.164
	30	0.118
	40	0.001
10	20	0.001
	30	0.000
	40	0.000

ตารางที่ 4.4 ค่า NMSE ของผลลัพธ์ที่ได้จากการระบบกับผลลัพธ์ที่ได้จากเครื่อง PC
(ข้อมูลตัวอย่างซึ่งได้จากการสัญญาณคลื่นเสียงพูดคำว่า "สาม")

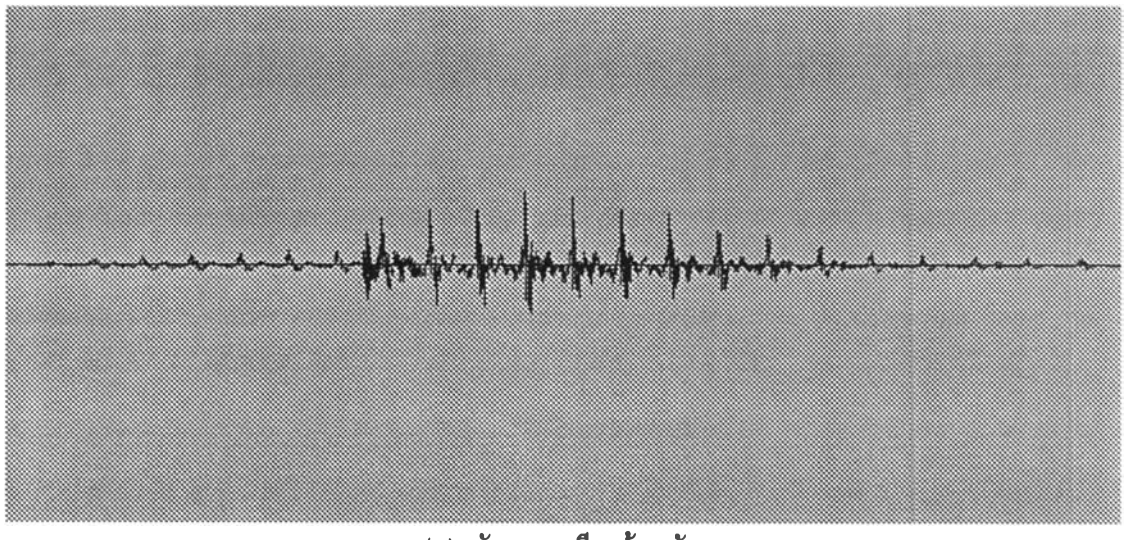
ลำดับที่ของ Frame ของสัญญาณเสียงพูด	ความยาวของ Frame (ms)	NMSE ($x10^{-3}$%)
1	20	0.072
	30	0.070
	40	0.068
5	20	0.060
	30	0.028
	40	0.001
10	20	0.000
	30	0.001
	40	0.001

ตารางที่ 4.5 ค่า NMSE ของผลลัพธ์ที่ได้จากการระบบกับผลลัพธ์ที่ได้จากเครื่อง PC
(ข้อมูลตัวอย่างซึ่งได้จากการสัญญาณคลื่นเสียงพูดคำว่า "สี่")

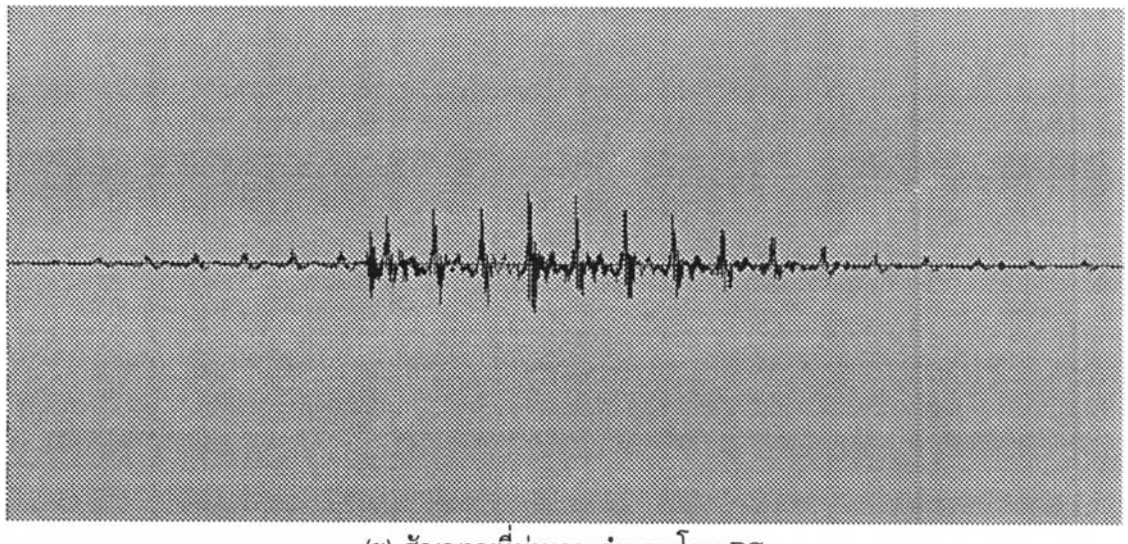
ลำดับที่ของ Frame ของสัญญาณเสียงพูด	ความยาวของ Frame (ms)	NMSE ($x10^{-3}$%)
1	20	0.408
	30	0.436
	40	0.406
5	20	0.262
	30	0.093
	40	0.044
10	20	0.066
	30	0.000
	40	0.001

จากรูปที่ 4.7 จะเห็นได้ว่าค่าที่ได้จากการคำนวณทั้งสองวิธีมีค่า NMSE ไม่เท่ากับ 0 นั้นแสดงว่าในการคำนวณมีความคลาดเคลื่อนเกิดขึ้นจึงทำให้ค่าที่ได้จากการคำนวณทั้งสองวิธีได้ผลลัพธ์ที่แตกต่างกัน และค่าความคลาดเคลื่อนที่เกิดขึ้นในตารางที่ 4.2 ถึง 4.5 มีค่าอยู่ในช่วงเดียวกันแสดงว่าความคลาดเคลื่อนในการคำนวณเกิดจากสาเหตุเดียวกันและสามารถอธิบายได้ดังนี้คือ ตัว MPU มีขนาดของริจีสเตอร์ขนาด 16 บิตและการจัดเก็บข้อมูลแบบ 16 บิต ดังนั้นในการคำนวณจึงได้ใช้การจัดการผลลัพธ์ของตัวประมวลผลแบบ Q Format [11] ซึ่งเป็นการเก็บค่า MSB จำนวน 16 บิตและจะไม่เก็บ LSB 16 บิตดังนั้นเมื่อมีการคำนวณทางด้านคณิตศาสตร์จะทำให้มีการผิดพลาดในการคำนวณสะสมแต่จากตารางจะเห็นว่ามีค่า MSE มีค่าไม่มากเนื่องจากโปรแกรมตัวอย่างซึ่งใช้ในการทดสอบมีความซับซ้อนของการคำนวณไม่มากและที่สำคัญไม่

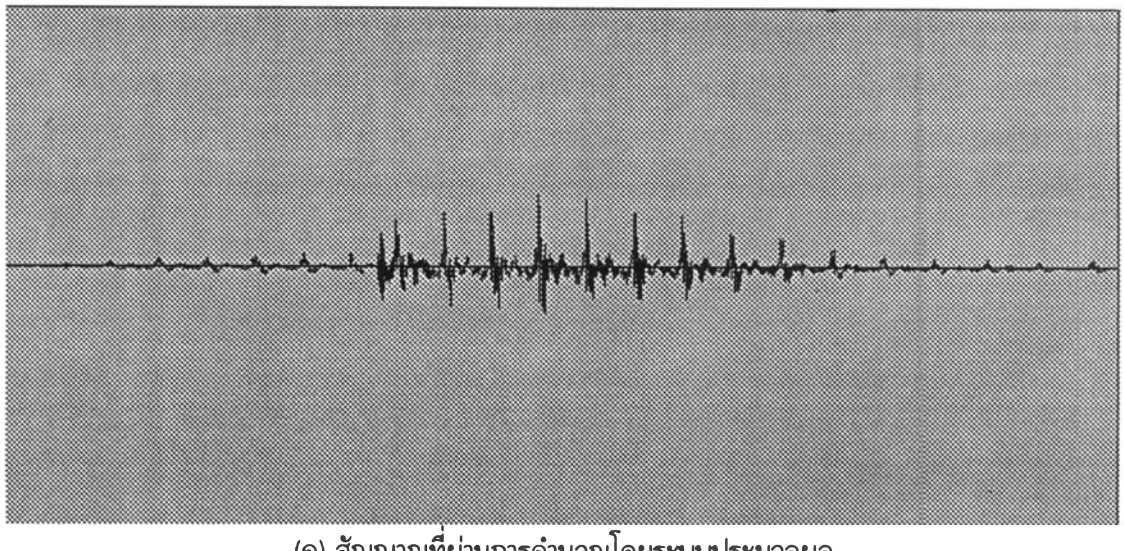
เป็นการคำนวณแบบเรียกตัวเอง (Recursive) และจากกราฟจะเห็นได้ว่าค่า NMSE จะมีค่าต่ำลงตามลำดับ Frame เพราะสัญญาณเสียงพูดในช่วงแรกมีขนาดของสัญญาณต่ำดังนั้นเมื่อเทียบกับความคลาดเคลื่อนที่ได้จากการคำนวณจะทำให้ค่า NMSE มีค่าสูงและสัญญาณเสียงพูดในช่วงกลางจะมีขนาดของสัญญาณสูงดังนั้นเมื่อเทียบกับความคลาดเคลื่อนที่ได้จากการคำนวณจะทำให้ค่า NMSE มีค่าต่ำลงและถึงแม้ว่าค่าที่ได้จากระบบประมวลผลจะมีค่าคลาดเคลื่อนไปบ้างแต่ค่าคลาดเคลื่อนมีค่าน้อย ค่า NMSE จะมีค่าเพิ่มขึ้นเมื่อมีลักษณะการคำนวณที่มีความซับซ้อนมากขึ้นดังนั้นการประยุกค์ใช้งานระบบประมวลผลจึงต้องคำนึงถึงความละเอียดที่ต้องการใช้งานด้วยและความละเอียดซึ่งได้ใช้การจัดการผลลัพธ์ของตัวประมวลผลแบบ Q Format [11] นั้นจะมีความละเอียดสูงขึ้นโดยขึ้นอยู่กับความเข้าใจกับอัลกอริทึมซึ่งจะนำมาประยุกต์ใช้



(ก) สัญญาณเสียงต้นฉบับ

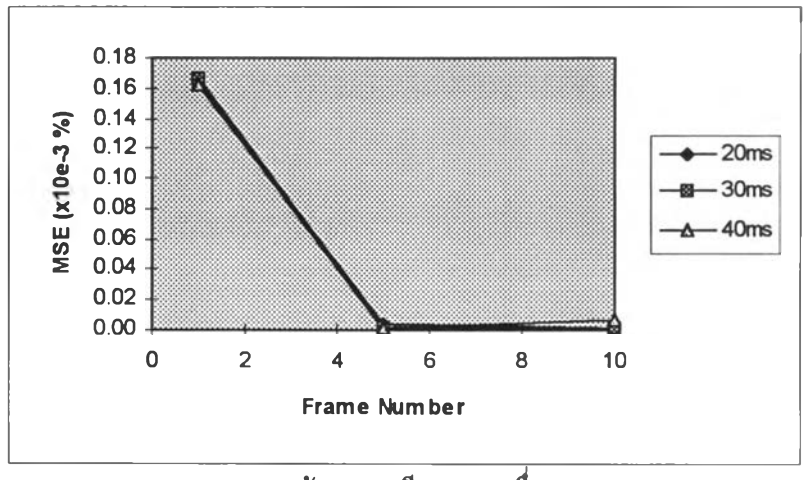


(ข) สัญญาณที่ผ่านการคำนวณโดย PC

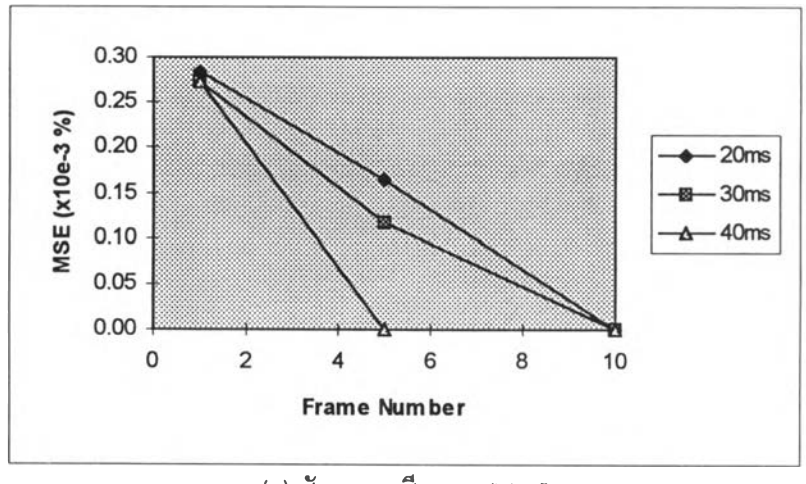


(ค) สัญญาณที่ผ่านการคำนวณโดยระบบประมวลผล

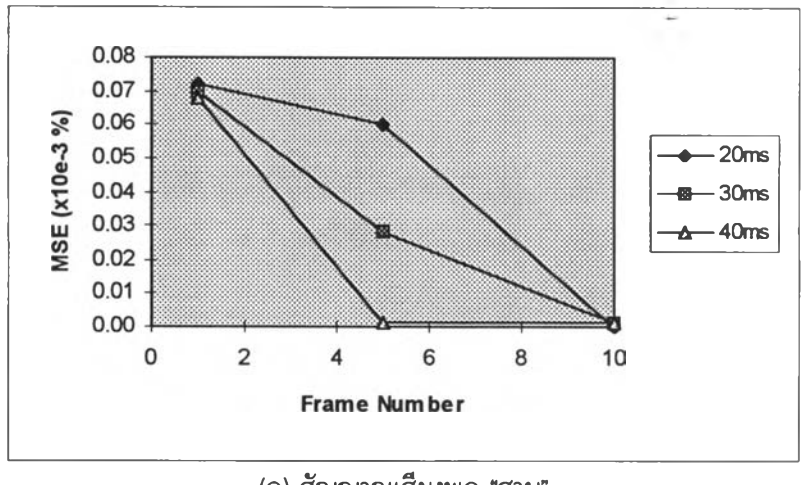
รูปที่ 4.6 เปรียบเทียบสัญญาณเสียงจากการคำนวณทั้ง 2 วิธีกับสัญญาณเสียงพูดต้นฉบับคำว่า "หนึ่ง"



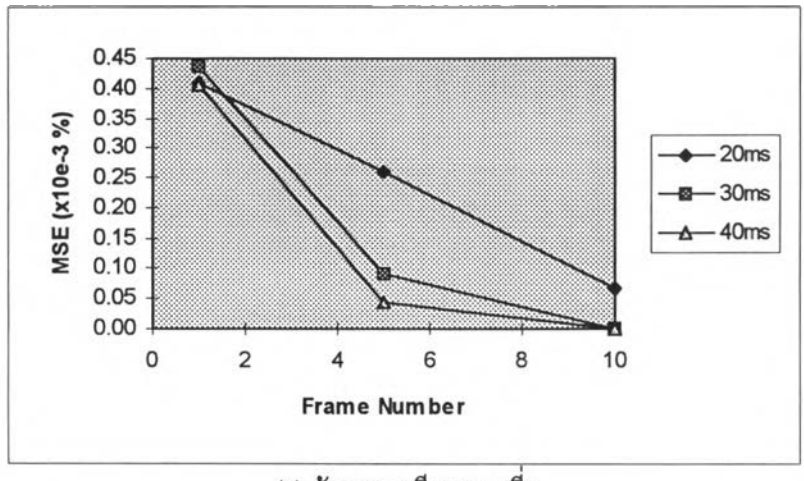
(ก) สัญญาณเสียงพูด "หนึ่ง"



(ข) สัญญาณเสียงพูด "สอง"



(ค) สัญญาณเสียงพูด "สาม"



(ง) สัญญาณเสียงพูด "สี่"

รูปที่ 4.7 กราฟเปรียบเทียบค่า NMSE ซึ่งคำนวณจากสัญญาณเสียงพูดต่างๆ

- ผลการเปรียบเทียบเวลาซึ่งใช้ในการประมวลผลระหว่างระบบประมวลผลกับ PC

ในการจับเวลาในการประมวลผลนั้นแบ่งได้เป็น 2 ส่วนดังนี้

1. เวลาซึ่งใช้ในการประมวลผลของเครื่อง PC (Pentium120)
2. เวลาซึ่งใช้ในการประมวลผลของระบบประมวลผล

เนื่องจากระบบประมวลผลใช้เวลาในการคำนวณซึ่งมีความเร็วสูงและยังมีลักษณะการทำงานแบบเวลาจริงดังนั้นจึงไม่สามารถทำการจับเวลาได้โดยตรงจึงต้องทำการหาจากการคำนวณจากชุดคำสั่งซึ่งได้ทำการ Download ลงในระบบประมวลผลซึ่งสามารถคำนวณได้ตามสมการที่ (4.3) [7]

$$time = \frac{(Default\ Clock) \times (cycle / Instruction) \times (time / cycle) \times (Number\ of\ Instruction)}{(Apply\ Clock)} \qquad (4.3)$$

โดย

- Default Clock : สัญญาณนาฬิกาซึ่งควรจะป้อนให้กับตัวประมวลผลเพื่อให้ตัวประมวลผลสามารถทำงานได้เต็มความสามารถ
- Apply Clock : สัญญาณนาฬิกาซึ่งป้อนให้กับตัวประมวลผลจริง
- cycle/Instruction : จำนวนรอบในการทำงานต่อหนึ่งคำสั่ง
- Number of Instruction : จำนวนคำสั่งที่ใช้ในการประมวลผล

จากสูตรที่ (4.3) จะสามารถทำการคำนวณเวลาได้ดังแสดงในตารางที่ 4.6

ตารางที่ 4.6 เวลาซึ่งใช้ในการประมวลผลสำหรับสัญญาณเสียงพูด
ระหว่างระบบกับเครื่อง PC (ในการคำนวณสัญญาณเสียงพูด 1 Frame หรือ 20 ms)

คำนวณด้วยระบบที่ออกแบบ (ms)	คำนวณด้วย PC (ms)
0.968	0.704

จากตารางที่ 4.6 จะเห็นได้ว่าเวลาที่ใช้ในการคำนวณโดยระบบประมวลผลใช้เวลาในการคำนวณมากกว่าเวลาที่ใช้โดย PC เนื่องจากสัญญาณนาฬิกาซึ่งป้อนให้กับตัวประมวลผลมีค่าเพียงครึ่งหนึ่งของสัญญาณนาฬิกาซึ่งสามารถป้อนให้กับตัวประมวลผลจึงทำให้ความเร็วในการประมวลผลลดลงครึ่งหนึ่งและตัวอย่างโปรแกรมซึ่งใช้ในการทดสอบระบบประมวลผลนั้นไม่ได้มีการทำการ Optimization เพื่อลดหรือรวมคำสั่งดังนั้นเวลาในการคำนวณจึงมากกว่าการคำนวณโดยเครื่อง PC แต่ถึงแม้ว่าระบบประมวลผลนั้นจะใช้เวลาในการคำนวณมากกว่าเครื่อง PC แต่เวลาในการคำนวณยังมีค่าน้อยกว่าเวลาในการรับเสียงพูดจำนวน 1 Frame หรือ 20 ms ซึ่งแสดงให้เห็นว่าระบบสามารถทำงานแบบเวลาจริงได้ และเวลาที่เครื่อง PC ใช้ในการคำนวณนั้นไม่ได้รวมถึงเวลาซึ่งใช้ในการรับและแปลงสัญญาณจากแอนะลอกมาเป็นสัญญาณดิจิตอลดังนั้นถึงแม้ว่าเครื่อง PC สามารถใช้เวลาในการคำนวณน้อยกว่าแต่เครื่อง PC ไม่สามารถทำงานแบบเวลาจริงได้