

國立台灣海洋大學資訊工程學系專題報告

題目

PIC16LF1826 UART/SPI modules

作者

學號:00957033 姓名:鄭鼎立 e-mail:leo4545525@gmail.com

學號:00957152 姓名:魏柏勝 e-mail:samwei9102@gmail.com

報告編號:NTUCSE 年度-嚴茂旭-小組編號

指導教授:嚴茂旭 博士

中華民國 112年 12 月 2 日

Project Report, NTOU CSE

Title

PIC16LF1826 UART/SPI modules

Author

00957033, 鄭鼎立, e-mail :leo4545525@gmail.com

00957152, 魏柏勝, e-mail :samwei9102@gmail.com

Project Report Number: XXX-XXX-XXXXX

Advisor : Dr. 嚴茂旭

2023 / 12 / 02

專題分工及貢獻度說明

編號	姓名	主要工作內容	專題貢獻度 (100%)
1	鄭鼎立	CPU 實作、 組合語言程式撰寫、 SPI 架構、 報告撰寫、 海報製作。	50%
2	魏柏勝	CPU 方塊圖及實作、 組合語言程式撰寫、 UART 實作、 報告撰寫。	50%

註：1. 主要工作內容：如工作分配、論文閱讀、系統實作、報告撰寫等(可多項)。

目錄

目錄.....	4
摘要:.....	5
一、 簡介:.....	6
1 研究動機:.....	6
2 研究目的:.....	6
二、 系統原理與架構實作:.....	7
1. 系統原理:.....	7
1.1 UART:	7
1.2 SPI:	8
2. 架構實作.....	9
2.1 UART 接收端	9
2.2 UART 傳送端	10
2.3 SPI 接收端	11
2.4 SPI 傳送端	13
2.5 CPU.....	15
2.6 組合語言程式撰寫	16
三、 實作結果.....	17
3.1 計算兩數之最大公因數	17
3.2 計算兩數之相加結果	18
四、 結論.....	19

摘要:

採用 PIC16LF1826 之 ISA，實作通用型非同步收發傳輸 UART(RS232)及同步序列傳輸 SPI，使連接的設備端，如個人電腦，達成對 MCU 微處理器的溝通。

本專案使用 polling 輪詢方式，將撰寫好的組合語言，如:計算最大公因數、兩數相乘、兩數相除，透過 CPU 經指令擷取後解碼運算並回傳個人電腦顯示結果，達成溝通之用意。

其中程式碼使用 SystemVerilog 撰寫、模擬波型圖使用 modelsim 觀看、並使用 MPLAB IDE 將組合語言轉成 HEX 編碼，再透過 ActivePerl 將 HEX 轉成可供 CPU 擷取之 ROM 格式。

一、簡介:

1 研究動機:

因曾修過計算機系統設計以及可重組式晶片系統設計。前者課程內容注重於撰寫 Verilog 建構 ISA，並把運算結果顯示在 FPGA 以便除錯；而後者注重如何將所定義的系統規格，做軟硬體的分割，且設計系統之軟硬體協同；因此有了將兩者之課程內容融合之動機。

2 研究目的:

為了讓 MCU 微處理器的周邊裝置與個人電腦溝通，須從電腦端輸入測試資料，接著在 FPGA 上即時顯示，以及傳回電腦端做顯示以取得 MCU 運算結果，以達成整個軟硬體流程控制，如圖 1所示。

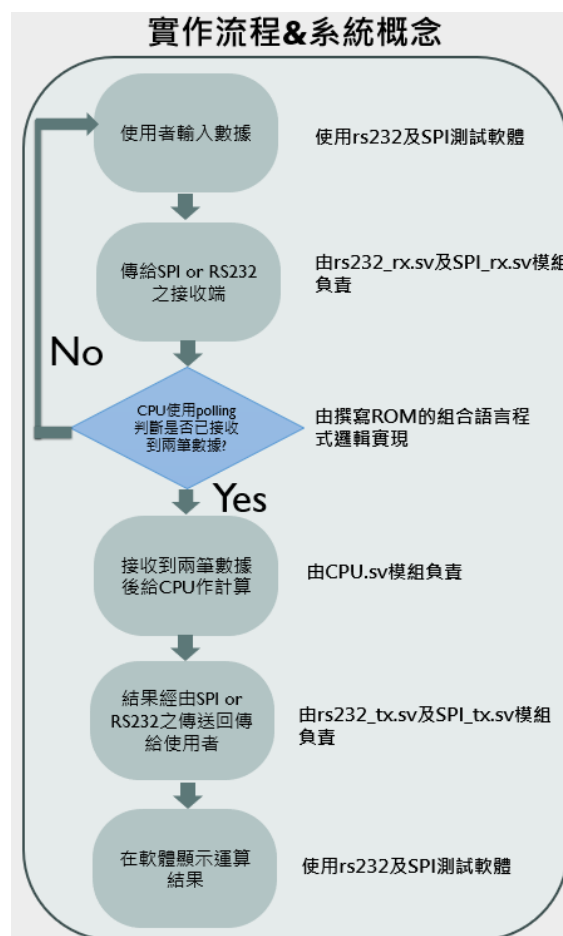


圖1、實作流程&系統概念

二、系統原理與架構實作:

1. 系統原理:

1.1 UART:

資料是以串列(Serial)方式由資料最低位元開始傳輸，其使用一個起始位元後面緊跟7或8個資料位元，然後是可選的奇偶校驗位元，最後是一或兩個停止位元，所以傳送一個字元至少需要10位元，如圖 2所示。

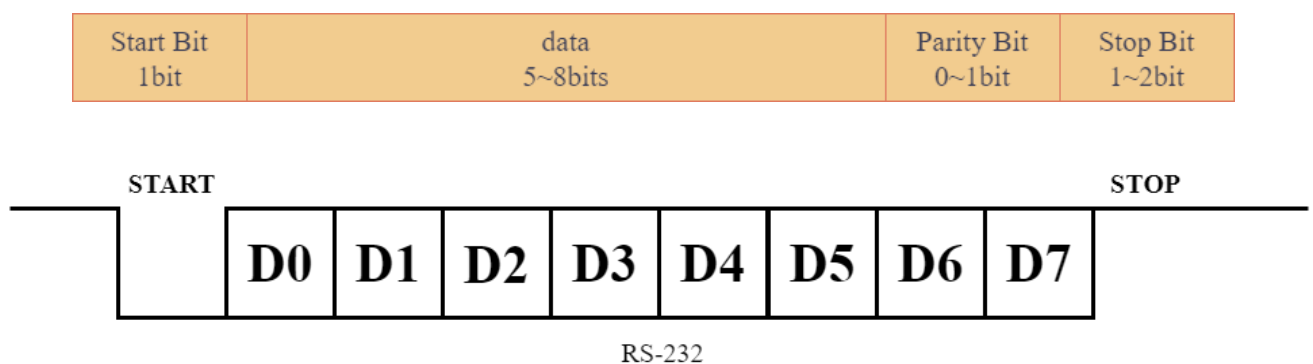


圖2、UART 封包格式

傳送端與接收端只需約定是以相同鮑率(Baud Rate)來傳輸；接收端的接收時脈(Receiver Clock, RxC)產生方式和傳送端的傳輸時脈(Transmitter Clock, TxC)無關，此種傳輸方式允許傳送與接收時脈的頻率有誤差，故稱為非同步(Asynchronous)傳輸，如圖 3所示。

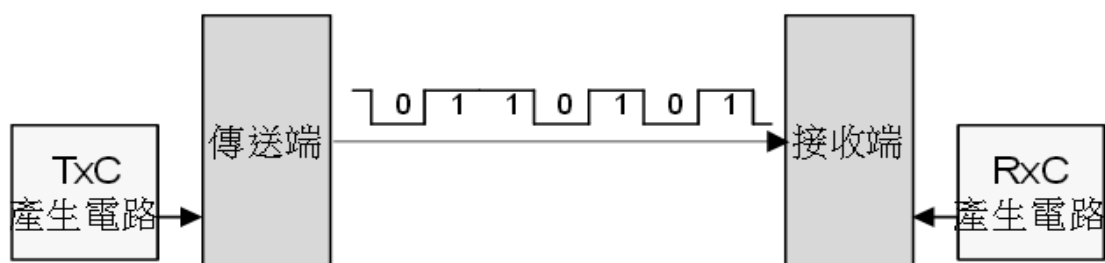


圖3、非同步傳輸示意圖

1.2 SPI:

是一個單主機多從機的主從式同步串列通訊，裝置之間使用全雙工模式通訊，線路上的訊號可以同時雙向傳送。接收端依據傳送端的時脈來接收資料，傳送端和接收端共用同一個時脈；傳送端以一條線送出資料，同時以另一條線送出傳送時脈，提供接收端之同步訊號，如圖 4所示。

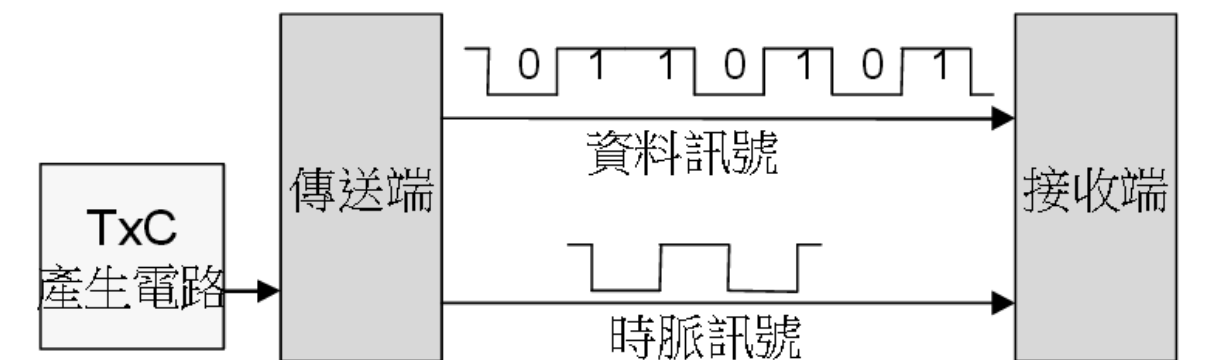


圖4、同步傳輸示意圖

主要腳位由

- MISO：slave 傳送資料到 master；master 接收從 slave 傳來的資料。
- MOSI：master 傳送資料到 slave；slave 接收從 master 傳來的資料。
- SCK：由 master 提供 clock 給 slave，使兩端同步。
- SSN：slave select，master 可選擇和特定 slave 傳輸資料。

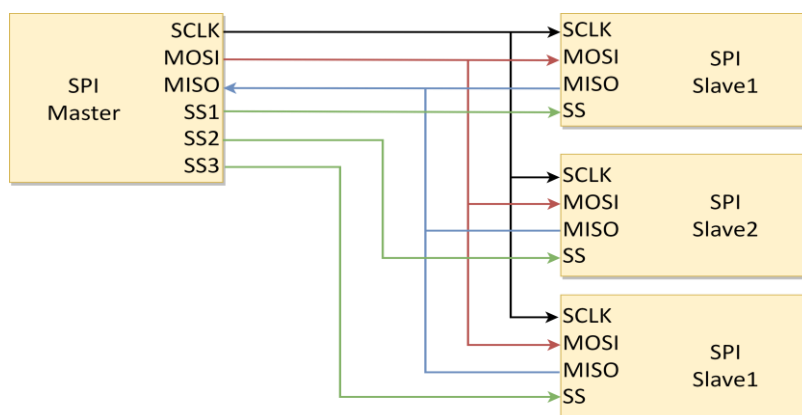


圖5、SPI 之主從關係圖

2. 架構實作

2.1 UART 接收端

使用者自個人電腦端輸入測試資料後，會先經由 UART 接收端接收資料；其中資料自最低位元開始傳輸，並將資料以 bit 為單位右移存入暫存器，如圖 6。

```
//shift register
always_ff@ (posedge clk) begin
    if(rst)
        shift_data[7:0] <= 8'b00000000;
    else if(bit_flag)
        shift_data[7:0] <= {rx, shift_data[7:1]};
end
```

圖6、UART 接收資料方式

接收端取得8bit 完整資料後，會通知 CPU 的特殊暫存器 PIR1，以進行後續運算，接收流程圖如圖 7。

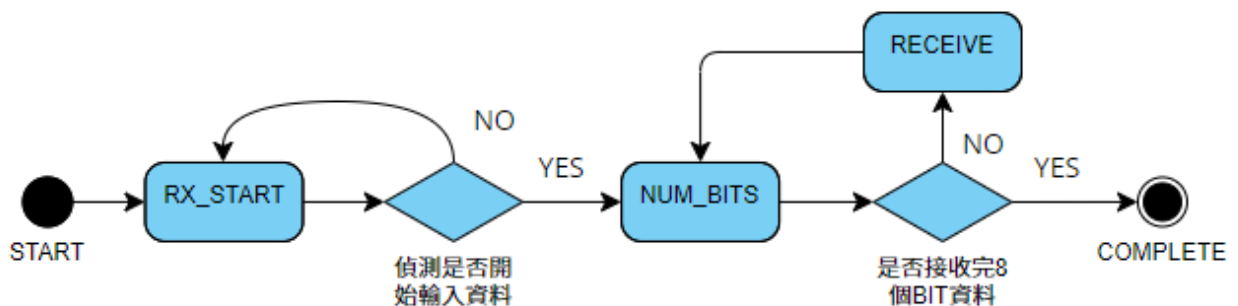


圖7、UART 接收資料流程圖

2.2 UART 傳送端

CPU 運算完畢後，會將結果存入特殊暫存器 TXREG。

而 UART 傳送端內部暫存器會接收來自 TXREG 的資料，並透過資料封裝回傳封包給個人電腦端，最後個人電腦端再自行解碼並將運算結果顯示在 UART 接收軟體介面。其中傳輸方向亦是從最低位元開始。

```
//data transfer
always_ff@ (posedge clk) begin
    if(rst)
        data <= 10'h3ff;
    else if(load_tx_data)
        data <= {1'b1, tx_data, 1'b0}; // {end_bit, data, start_bit}
    else if(bit_flag)
        data <= {1'b1, data[9:1]};
end
```

圖8、UART 封裝資料及傳輸方式

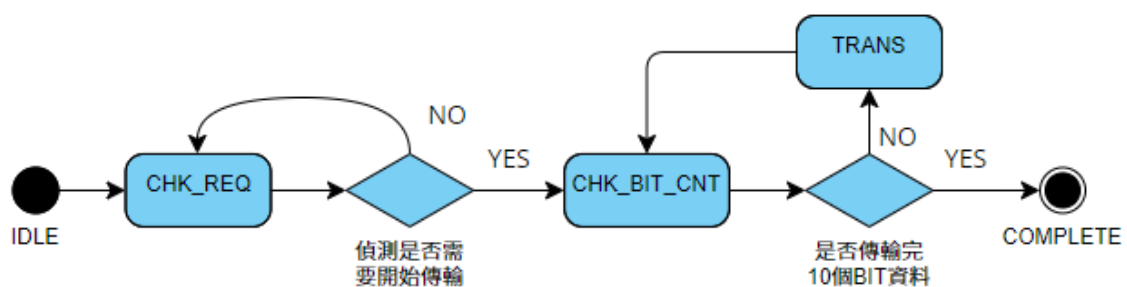


圖9、UART 傳輸資料流程圖

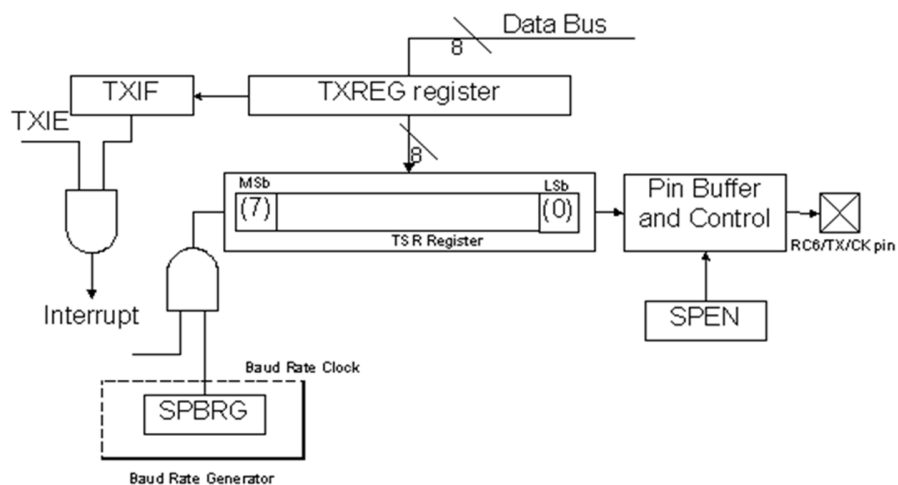


圖10、UART 傳送端方塊圖

2.3 SPI 接收端

使用者輸入資料，會觸發 `ssn_neg` 接腳，把資料傳給 CPU 的 SSP1BUF 暫存器，傳輸方向從 MSB 開始傳輸，見圖 11、圖 12、圖 13。

```
always_ff@(posedge clk_100MHz or posedge reset) //移位用cnt
begin
    if      (reset)          shi_data <= 0;
    else if (sclk_pos)       shi_data <= {shi_data[30:0], mosi}; //因為傳輸方向是由MSB開始傳輸,所以做左移
end
```

圖11、SPI_rx 移位邏輯

```
case(fsm_ps)
    INIT:
    begin
        fsm_ns = START_SPI_RX;
    end
    START_SPI_RX:
    begin
        if(ssn_neg)
            fsm_ns = REC_PKG;
        end
    end
    REC_PKG:
    begin
        if(rec_data_cnt == 32)
            fsm_ns = FINISH_WRITE;
        end
    end
    FINISH_WRITE:
    begin
        rec_data_cnt_rst = 1;
        rx_finish        = 1;
        fsm_ns           = START_SPI_RX;
    end
endcase
```

圖12、SPI 接收端之組合邏輯

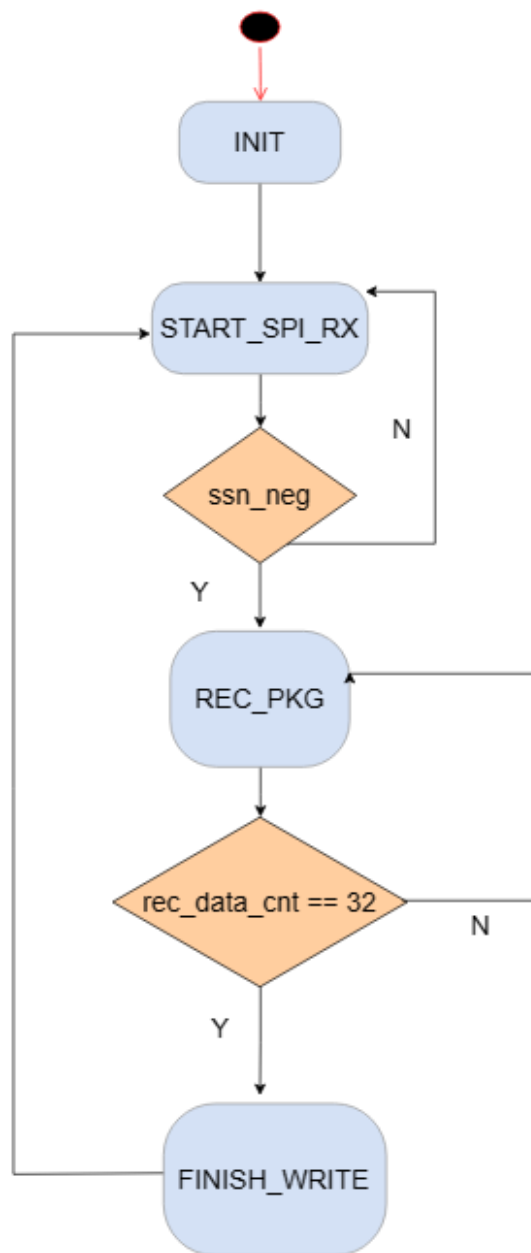


圖13、SPI 接收端之狀態圖

2.4 SPI 傳送端

輸入腳位 data_read 會接收來自 CPU 計算好的資料，並且透過此模組回傳給軟體顯示計算好的結果，見圖 14、圖 15、圖 16。

```
always_ff@(posedge clk_100MHz or posedge reset) //移位用cnt
begin
    if      (reset)          miso      <= 0;
    else if (sclk_neg)        miso      <= shi_data[15];
end

always_ff@(posedge clk_100MHz or posedge reset) //移位用cnt
begin
    if      (reset)          shi_data <= 0;
    else if (sclk_neg)        shi_data <= {shi_data[14:0], 1'b0}; //因為傳輸方向是由MSB開始傳輸,所以做左移
    else if (load_shi_data)    shi_data <= data_read;
end
```

圖14、SPI_tx 移位邏輯

```
case(fsm_ps)
    INIT:
    begin
        fsm_ns = START_SPI_TX;
    end
    START_SPI_TX:
    begin
        if(tx_req)
            fsm_ns = SEND_DATA;
        end
    SEND_DATA:
    begin
        send_data_cnt_rst = 1;
        load_shi_data = 1;
        fsm_ns = SHIFT;
    end
    SHIFT:
    begin
        if(send_data_cnt >= 16)
            fsm_ns = FINISH;
        end
    FINISH:
    begin
        send_data_cnt_rst = 1;
        fsm_ns = INIT;
    end
endcase
```

圖15、SPI 傳送端之組合邏輯

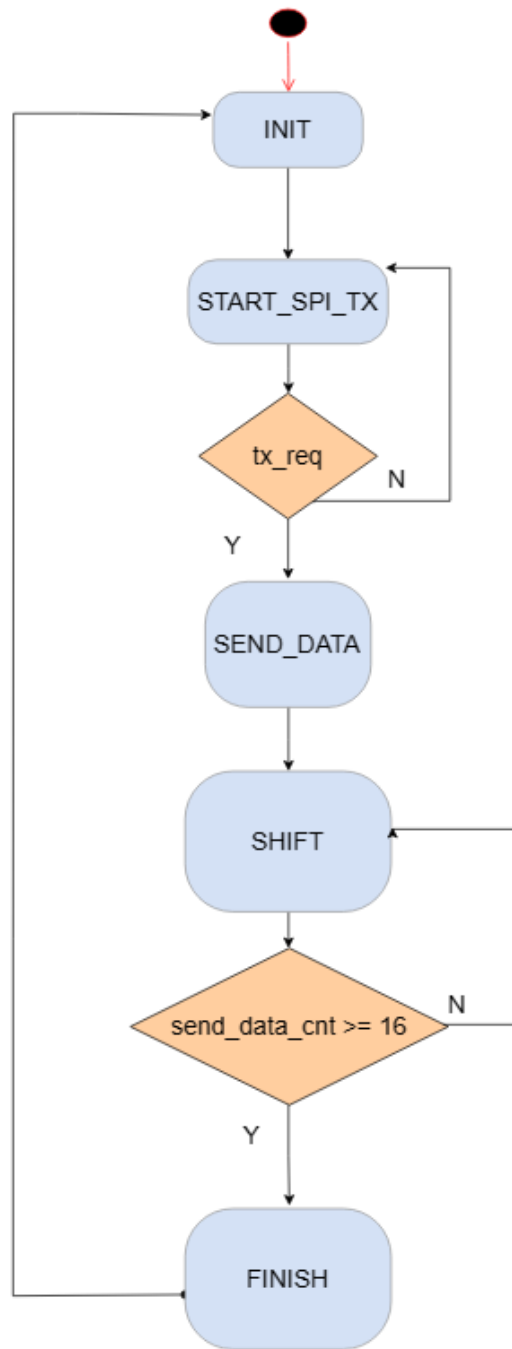


圖16、SPI 傳送端之狀態圖

2.5 CPU

依照 PIC16LF1826 規格書實作，將 PC 的值對應到 MAR，再利用 MAR 的值取得指令記憶體位置，接著 CPU 將指令做解碼，並由控制單元決定該由哪些功能單元負責運算或傳遞資料。

註:PIR1[4]為 TXIF、PIR1[5]為 RCIF

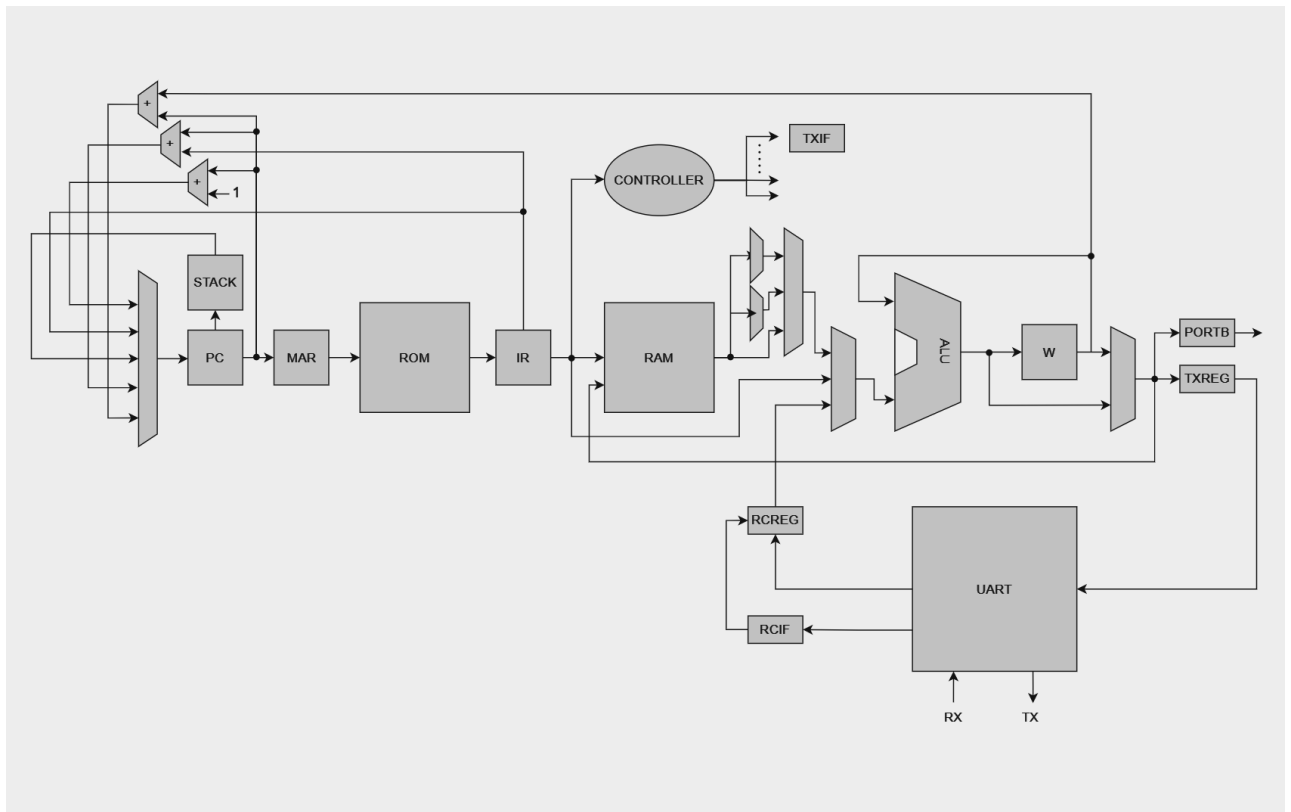


圖17、CPU 架構圖

```
//instruction set
logic ADDWF, ANDWF, CLRF, CLRW, COMF, DECF, GOTO, MOVLW, ADDLW, IORLW, ANDLW, SUBLW, XORLW
logic INCF, IORWF, MOVF, MOVWF, SUBWF, XORWF
logic ASRF, LSLF, LSRF, RLF, RRF, SWAPF
logic BCF, BSF, BTFSC, BTFSS, DECFSZ, INCFSZ
logic BRA, BRW, NOP
logic CALL, RETURN
```

圖18、此架構可支援之指令集

2.6 組合語言程式撰寫

以 UART 傳輸介面實作「兩數相加程式」為例。

PIR1: CPU 內部之狀態暫存器，若 RCREG 完成接收則 PIR[5]為1。

BTFSS: 偵測 PIR1[5]，若 PIR1[5]為1，則會跳過下一行指令 GOTO。

MOVF: RCREG 之資料存入 W 暫存器，達成輪詢之目的。

MOVWF: W 暫存器之資料存入 TXREG，並通知 UART 傳送端。

```
#include    <p16LF1826.inc>

num1       equ    0x25

;*****
;               Program start                *
;*****

                org    0x00                ; reset vector

init           clrfs    num1                ; ram[25]=0
                clrw                    ; w=0

polling1       btfss    PIR1, 0x05
                goto     polling1
                movf     RCREG, 0          ; w=RCREG
                addwf    num1, 1          ; ram[25]=ram[25]+w

polling2       btfss    PIR1, 0x05
                goto     polling2
                movf     RCREG, 0          ; w=RCREG
                addwf    num1, 0          ; w=ram[25]+w
                movwf    TXREG

                goto     init
end
```

圖19、組合語言範例

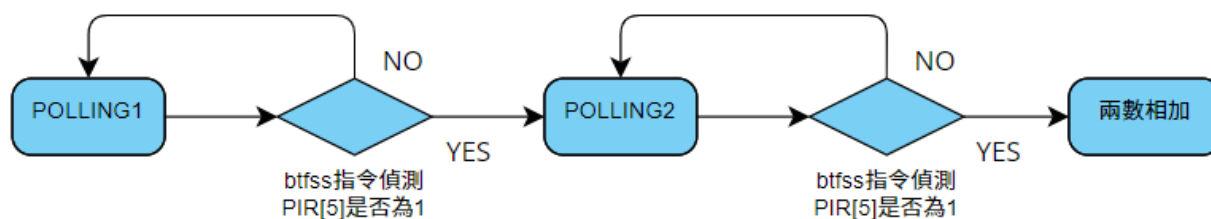


圖20、本範例流程圖

三、實作結果

3.1 計算兩數之最大公因數

以 UART 傳輸介面實作「兩數之最大公因數程式」為例，
能看到正確計算

$\text{gcd}(60, 48) = 12$ 、 $\text{gcd}(30, 24) = 6$ 、 $\text{gcd}(4, 2) = 2$ 、 $\text{gcd}(18, 11) = 1$ 、
 $\text{gcd}(200, 50) = 50$ 。

且 UART 傳送端內部暫存器正確接收來自於 TXREG 的資料，再經由 tx 接腳回傳
個人電腦端。



圖21、使用 modelsim 觀看結果之波型圖

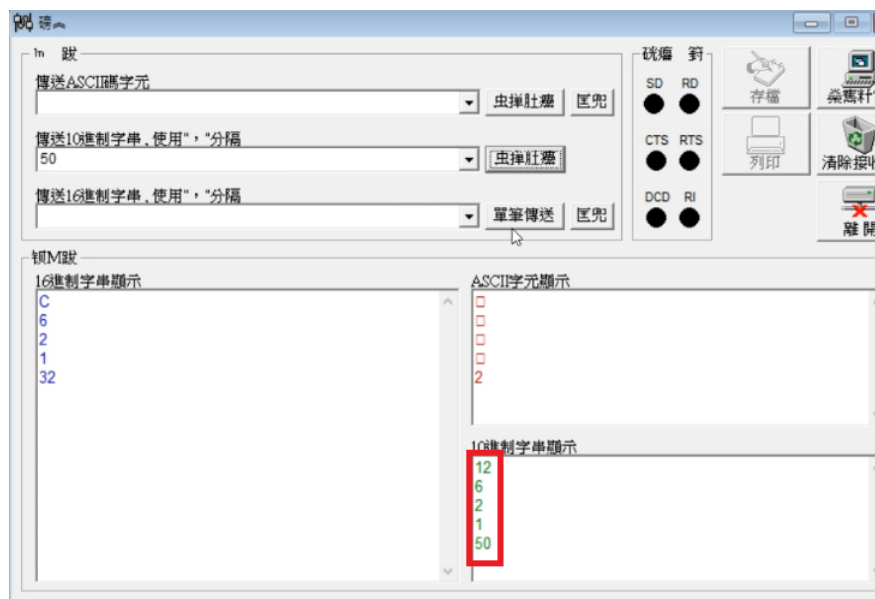


圖22、使用 UART 測試軟體測試結果

3.2 計算兩數之相加結果

以 SPI 傳輸介面實作計算兩數之相加結果程式為例，並使用 modelsim 軟體觀看模擬波形。能看到正確計算 $5+4=9$ 、 $20+30=50$ （16進制:32）、 $10+15=25$ （16進制:19）。

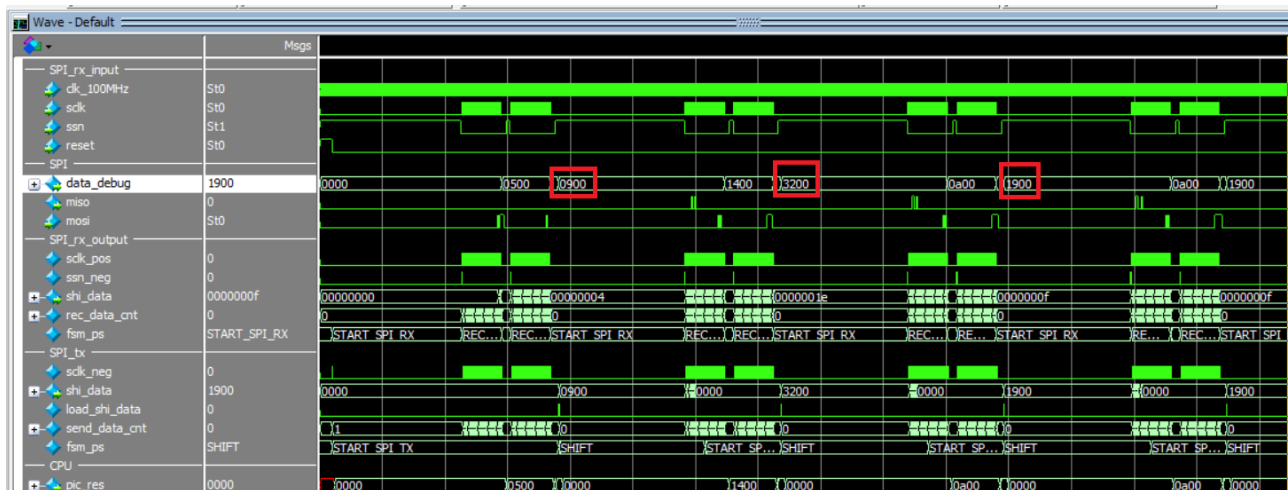


圖23、使用 modelsim 觀看結果之波型圖

四、結論

本專案目的為創造一個方便客制化的系統，讓使用者能依照需求撰寫程式，以替換 ROM 之內容，即能實現不同運算需求。因為此模組能嵌入其他系統，如資料加密等，充分展現可擴充性。而採用輪詢方法取得資料，優點在於針對 I/O 密集型系統、資料負載量大的系統，能比採用中斷方法付出更少的成本，以達成相同效益。