

Các TD lập trình Z80 (phần đọc thêm của VXL-chương 2)

Chú ý:

Các TD sau được chạy thử trên mô phỏng Z80 (Z80 IDE V1.0 của Peter J.Fondse). Trong mô phỏng này, tác giả đã định nghĩa sẵn các ngoại vi như sau (ta có thể chỉnh lại ở cửa sổ Z80 Visual Simulator → Peripherals → Configure Peripherals)

STT	Ngoại vi	Loại Cổng	Địa chỉ	Chú thích
1	Các công tắc (Switches)	Nhập	02H	
2	8 đèn LED (LED's)	Xuất	03H	0=LED tắt; 1=LED sáng
3	Thanh trượt (Slider)	Nhập	04H	
4	Hiển thị vạch (Bar display)	Xuất	05H	
5	Hiển thị 7 đoạn (7 segment display)	Xuất	0AH	4 digit với dấu chấm thập phân

TD 1: Các thí dụ cộng/trừ số nhị phân và cộng BCD

; TD: binary numbers ex1

; Minh họa các phép toán nhị phân

NUM1 EQU 25

NUM2 EQU -17

BIN16_1 EQU 2578 ; = 0A12H

BIN16_2 EQU 1289 ; = 0509H

TEST EQU 1289-2578 ; = FAF7H

org 100h

program

IN A,(02H) ; đọc từ switches

OUT (03H),A; xuất ra LED

CPL ; lấy bù 1 của A

OUT (03H),A; xuất ra LED bù 1 của A

CPL ; 2 lần bù 1 về trị cũ

OUT (03H),A; xuất ra LED

XOR 0FFH ; 1 cách lấy bù 1 khác

OUT (03H),A; xuất ra LED bù 1 của A

XOR 0FFH ; 2 lần bù 1 về trị cũ

NEG ; bù 2 của A

OUT (03H),A; xuất ra LED bù 2 của A

NEG ; 2 lần bù 2 về trị cũ

; Thu lệnh ADD với toán hạng 8 bit

LD A,NUM1

LD B,NUM2

ADD A,B ; A <- A + B = 25 + (-17) = 8

OUT (03H),A; xuất ra LED

; Thu lệnh SUB với toán hạng 8 bit

LD A,NUM1

LD B,NUM2

SUB B ; A <- A - B = 25 - (-17) = 42

OUT (03H),A; xuất ra LED

; Thu lệnh ADD với toán hạng 16 bit dùng cặp thanh ghi HL, DE

LD DE,BIN16_1

LD HL,BIN16_2

ADD HL,DE ; HL <- HL + DE = 1289 + 2578 = 3867 = 0F1BH

; Thu lệnh SBC với toán hạng 16 bit dùng cặp thanh ghi HL, DE

SCF

CCF ; Xóa cờ Carry (CF=0) hoặc có thể dùng 1 lệnh AND A

LD DE,BIN16_1

LD HL,BIN16_2

SBC HL,DE ; HL <- HL - DE = 1289 - 2578 = -1289 = FAF7H

; Thu lenh ADD voi toan hang 16 bit dung thanh ghi 8 bit

LD IX,BIN16_3 ; Lay byte thap truoc

LD A,(IX)

LD IY,BIN16_4

LD B,(IY)

ADD A,B

LD DE,RESULT_1

LD (DE),A

INC IX ; Pointer chi den byte cao

INC IY

INC DE

LD A,(IX)

LD B,(IY)

ADC A,B

LD (DE),A

; Lenh SBC cung lam tuong tu!!

LD IX,BIN16_3 ; Lay byte thap truoc

LD A,(IX)

LD IY,BIN16_4

LD B,(IY)

SUB B

LD DE,RESULT_2

LD (DE),A

INC IX ; Pointer chi den byte cao

INC IY

INC DE

LD A,(IX)

LD B,(IY)

SBC A,B

LD (DE),A

; Cong 2 so BCD 4 digits

LD IX,BIN16_5 ; Lay byte thap truoc

LD A,(IX)

LD IY,BIN16_6

LD B,(IY)

ADD A,B

DAA

LD DE,RESULT_3

LD (DE),A

INC IX ; Pointer chi den byte cao

INC IY

INC DE

LD A,(IX)

LD B,(IY)

ADC A,B

DAA

LD (DE),A

HALT

BIN16_3 DEFW 1289

BIN16_4 DEFW 2578

BIN16_5 DEFW 1289H; Bieu dien BCD cua 1289

BIN16_6 DEFW 2578H; Bieu dien BCD cua 2578

RESULT_1 DEFS 2

RESULT_2 DEFS 2

RESULT_3 DEFS 2

end program

TD 2: Tính tổng của 1 mảng dữ liệu trong bộ nhớ

org 100H

; Chương trình tính tổng 1 array với số bytes (trong B) và với địa chỉ bắt đầu

; trong HL

```

; Ket qua duoc cat trong DE
program
    LD A,(N)
    LD B,A      ; Lay so bytes dat vao B
    LD HL, DATA
    LD A, 0
    LD DE,0     ; Sum <- 0
LOOP:    ADD A,(HL)
    LD E,A      ; Cap nhat byte thap cua Sum
    JR NC, NEXT
    INC D       ; Cap nhat byte cao cua Sum
NEXT:    INC HL
    DJNZ LOOP
    HALT
DATA:    DEFB 10
    defb 20
    defb 30
    defb 40
    defb 50
    defb 60
    defb 70
    defb 80
N:       defb $-DATA ; $ la dia chi hien tai
           ; do $ - DATA = 8
    end program

```

TD 3: Thử lệnh IN/OUT

```

; TD: LED ex1
; cac LED o cong xuat (03H) se tat/sang theo trang thai cua khoa o
; cong nhap (02H)
    org 100h
program
loop:    in A,(02H)
    out (03H),A
    jr loop
    end program

```

TD 4: LED sáng lần lượt từ phải sang trái và lặp lại

```

; TD: LED ex2 - LED sang chay xoay sang trai
PATTERN EQU 01H
    org 100h
program
    LD C,PATTERN
loop:    LD A,C
    OUT (03H),A
    RLC C ; Dung RRC C se chay sang phai!
    CALL DELAY
    JR loop
; Chuong trinh con DELAY lam tre
DELAY:   LD B,0 ; vong lap voi B la 256 lan!
L1:      LD D,0 ; vong lap voi D la 256 lan!
L2:      DEC D
    JR NZ,L2
    DJNZ L1
    RET
    end program

```

TD 5: Điều khiển LED sáng chạy sang trái/phải theo trị số 0/1 ở công tắc 0

```

; TD: LED ex3 - LED sang chay xoay sang trai/phai theo cong tac 0 la 0/1
PATTERN EQU 01H
    org 100h

```

```

program
    LD C,PATTERN
LOOP:   IN A,(02H)
        BIT 0,A
        JR NZ,RIGHT
        RLC C ; Dung RRC C se chay sang phai!
        JR NEXT
RIGHT:  RRC C
NEXT:   CALL DELAY
        LD A,C
        OUT (03H),A
        JR LOOP
; Chuong trinh con DELAY lam tre
DELAY:  LD B,0 ; vong lap voi B la 256 lan!
L1:     LD D,0 ; vong lap voi D la 256 lan!
L2:     DEC D
        JR NZ,L2
        DJNZ L1
        RET
end program

```

TD 6: Hiển thị các số từ 0 đến 9 và lặp lại

Các LED 7 đoạn trong hiển thị 4 ký số lần lượt có các địa chỉ là 0AH, 0BH, 0CH, và 0DH cho các ký số tương ứng từ vị trí cao xuống thấp. Trong TD này ta xuất LED 7 đoạn ở vị trí tận cùng bên phải, nên dùng địa chỉ 0DH!

; TD: LED ex4 - Hien thi cac so tu 0 den 9 va lap lai
org 100h

```

program
    LD L,0
LOOP:   LD A,L
        CALL DEC2LED
        OUT (0DH),A ; Xuat ra LED 7 doan tai digit don vi!
        CALL DELAY
        INC L
        LD A,9
        CP L
        JR NC,LOOP
        LD L,0
        JR LOOP
; Chuong trinh con DELAY lam tre
DELAY:  LD B,0 ; vong lap voi B la 256 lan!
L1:     LD C,100; vong lap voi C la 100 lan!
L2:     DEC C
        JR NZ,L2
        DJNZ L1
        RET
; Chuong trinh doi so nhi phan 4 bit sang ma 7 doan
DEC2LED: PUSH DE
        PUSH IX
        AND 0FH
        LD E,A
        LD D,0
        LD IX,XLAT_TAB
        ADD IX,DE
        LD A,(IX+0)
        POP IX
        POP DE
        RET
XLAT_TAB:
; LED 7 doan trong mo phong nay co thu tu cac doan nhu sau
;          hgfedcba - 0=LED tat, 1=LED sang

```

```

DEFB 00111111B; 0
DEFB 00000110B; 1
DEFB 01011011B; 2
DEFB 01001111B; 3
DEFB 01100110B; 4
DEFB 01101101B; 5
DEFB 01111101B; 6
DEFB 00000111B; 7
DEFB 01111111B; 8
DEFB 01101111B; 9
end program

```

TD 7: Tìm bit kiểm tra chẵn lẻ cho 1 số 8 bit trong thanh ghi A.

Trong Z80 có hỗ trợ việc cập kiểm tra chẵn lẻ với cờ P/V ở bit thứ 2 trong thanh ghi F sau các lệnh logic như AND, OR,... Tuy nhiên trong phần mềm mô phỏng của tác giả Peter J.Fondse không có thực hiện chức năng này, do đó ta sẽ tự viết chương trình cập nhật P/V theo nội dung của thanh ghi A: nếu số bit 1 ở A là số chẵn thì P/V=1 (điều kiện PE đúng), ngược lại thì P/V=0 (điều kiện PO đúng)

```

org 100H
program
; Hieu ung cap nhat co Parity sau lenh LOGIC
LD A,1
AND 0FFH ;P/V=0=Odd Parity(do so bit 1 cua ket qua AND la le)
LD A,3
AND 0FFH ;P/V=1=Even Parity(do so bit 1 cua ket qua AND la chan)
; Tuy nhien co P/V khong cap nhat vi Simulator nay bi loi trong phan cap nhat
; Parity. Neu dung phan mo phong Z80 Simulator IDE cua Oshon thi khong can
; Do do ta se 1 chuong trinh con se cap nhat Parity theo so bit 1 trong A
LD A,1
CALL PARITY_E ; P/V=0
LD A,3
CALL PARITY_E ; P/V=1
HALT
PARITY_E:
PUSH BC
PUSH DE
PUSH AF
POP BC ; B = A , C= F
LD B,8
LD D,A
LD E,0
RES 2,C ; P/V = 0 (co P/V o vi tri bit 2 trong thanh ghi F)
LOOP: SRL D
JR NC,SKIP
INC E
SKIP: DJNZ LOOP ; Vong lap tren dem so bit 1 trong A va cat vao E
BIT 0,E
JR NZ,ODD
SET 2,C ; P/V = 1
ODD: LD B,A
PUSH BC
POP AF ; A = khong doi, F duoc cap nhat theo so bit 1 cua A
POP DE
POP BC
RET
end program

```

TD 8: Áp dụng lệnh dịch bit để thực hiện phép nhân không dấu X cho với 1 số có dạng $2^n + 1$ (như 3, 5, 9, 17...): $X * (2^n + 1) = X*2^n + X$ = dịch X sang trái n bit + X

Giả sử kết quả của phép nhân sẽ đặt trong 1 số 16 bit.

→ Chương trình con MUL_2NA1 nhân 1 số ở thanh ghi A với 1 số có dạng $2^n + 1$ ở B và kết quả 16 bit được để trong cặp thanh ghi DE

```

    org 100H
program
    LD A,27
    LD C,17
    CALL MUL_2NA1
    HALT
MUL_2NA1:
    PUSH AF
    PUSH BC
    LD D,0
    LD E,A
    DEC C
    RET Z
    CALL SHIFT_N
    ADD A,E
    LD E,A
    LD A,0
    ADC A,D
    POP BC
    POP AF
    RET
; Tính DE = A x 2^n bằng cách dịch DE sang trái n bit (đặt trong C)
SHIFT_N:
    SRL C
    RET C
    SLA E ; ; Dịch cặp thanh ghi DE sang trái 1 bit
    RL D
    JR SHIFT_N
end program

```

Chú ý: Tóm tắt hiệu ứng của các lệnh dịch và xoay bit trong Z80

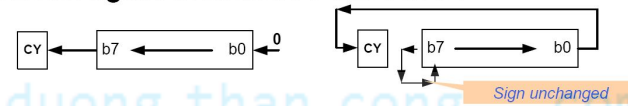
■ Rotate Left/Right Circular : RLC/RRC



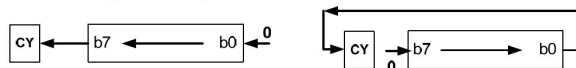
■ Rotate Left/Right : RL/RR



■ Shift Left/Right Arithmetic : SLA/SRA



■ Shift Left/Right Logical : SRL (no SLL = SLA)



TD 9: Tính bù 2 của 1 số 16 bit trong bộ nhớ

```

    org 100H
program
; Lay bu 2 cua 1 so 16 bit NUM1 trong bo nho o dia chi 200H
; va cat lai vao bien 16 bit NUM2 trong bo nho o dia chi 202H
    LD HL,(NUM1)    ; Neu (NUM1) = 56D3H
    LD A,L
    CPL              ; tinh bu 1 cua byte thap
    LD L,A
    LD A,H

```

```

        CPL                ; tính bù 1 của byte cao
        LD H,A
        INC HL
        LD (NUM2),HL ; thì bù 2 của nó (NUM2) = A92DH
; Cach 2 lấy bù 2 bằng lấy 0 - NUM1
        LD HL,(NUM1)
        EX DE,HL
        LD HL,0
        AND A            ; CF =0
        SBC HL,DE
        LD (NUM3),HL
        HALT
        org 200H
NUM1    defw 56D3H
NUM2    defs 2 ; bù 2 của 56D3H là A92DH
NUM3    defs 2
        end program

```

TD 10: Minh họa chuyển đổi biểu diễn số theo ASCII sang BCD và ngược lại; và tính tổng 2 số BCD có nhiều ký số, trong TD này mỗi số BCD có 10 ký số.

```

        org 100H
BCD_LEN EQU 5
program

        LD HL,NUM1
        LD IX,NUM1_BCD
        CALL ASCII_HEX
        LD HL,NUM2
        LD IX,NUM2_BCD
        CALL ASCII_HEX
        LD HL,NUM1_BCD + BCD_LEN - 1 ; chỉ đến ký số BCD tận cùng phải
        LD IX,NUM2_BCD + BCD_LEN - 1
        LD IY,BCD_SUM + BCD_LEN - 1
        CALL SUM_BCD
        LD HL,BCD_SUM
        LD IX,ASCII_SUM
        CALL BCD_ASCII
        HALT
ASCII_HEX:
        LD B,BCD_LEN
LOOP:    LD A,(HL)
        AND 0FH
        RLA
        RLA
        RLA
        LD C,A
        INC HL
        LD A,(HL)
        AND 0FH
        ADD A,C
        LD (IX+0),A
        INC HL
        INC IX
        DJNZ LOOP
        RET
SUM_BCD:
        LD B,BCD_LEN
        AND A ; CF=0
L1:     LD A,(IX+0)
        ADC A,(HL)
        DAA
        LD (IY+0),A

```

```

        DEC IX
        DEC IY
        DEC HL
        DJNZ L1
        RET
BCD_ASCII:
        LD B,BCD_LEN
L2:      LD A,(HL)
        LD C,A
        AND 0F0H
        RRA
        RRA
        RRA
        RRA          ; lay nua byte cao
        OR 30H        ; Doi sang ASCII tuong duong
        LD (IX+0),A
        INC IX
        LD A,C
        AND 0FH        ; lay nua byte thap
        OR 30H        ; Doi sang ASCII tuong duong
        LD (IX+0),A
        INC IX
        INC HL
        DJNZ L2
        RET
org 200H
NUM1      DEFM '1234567890'
NUM2      DEFM '1098765432'
ASCII_SUM DEFS 10 ; Ket qua la '2333333322'
          DEFB 0
NUM1_BCD  DEFS 5 ; Ket qua la 1234567890H (BCD cua thap phan 1234567890)
NUM2_BCD  DEFS 5 ; Ket qua la 1098765432H (BCD cua thap phan 1098765432)
BCD_SUM   DEFS 5 ; Ket qua la 2333333322H
end program

```

TD 11: Tìm chiều dài của 1 chuỗi ký tự

Chuỗi ký tự bắt đầu từ ô nhớ 0201H và đánh dấu hết chuỗi bằng ký tự “carriage return) (CR) có giá trị là 0DH. Đặt chiều dài chuỗi (không kể CR) vào ô nhớ 0200H.

TD: Từ ô nhớ 0201H ta thấy các byte: 52H, 41H, 54H, 48H, 45H, 52H, 0DH

Nghĩa là chuỗi ‘RATHER’, 0D : ‘R’, ‘A’, ‘T’, ‘H’, ‘E’, ‘R’, CR

Sau khi thực thi đoạn chương trình này tại ô nhớ 0200H có trị 06H.

→ Cách 1:

```

LD      HL, 0201H    ; Contrô HL chỉ đến đầu chuỗi
LD      B,0          ; Chiều dài chuỗi = B = 0
LD      A, 0DH       ; So sánh CR với ký tự trong chuỗi
CHKCR:  CP           (HL) ; Ký tự đang xét là CR?
        JR Z,DONE     ; Đúng. Kết thúc.
        INC B         ; Cộng 1 vào chiều dài chuỗi
        INC HL
        JR CHKCR      ; Kiểm tra ký tự kế
DONE:   LD A,B        ; Cất chiều dài chuỗi
        LD (0200H),A
        HALT

```

→ Cách 2: Dùng CPIR

```

LD      HL, 0201H    ; Contrô HL chỉ đến đầu chuỗi
LD      BC,0         ; Bộ đếm số byte BC = 0 dùng cho CPIR
LD      A, 0DH       ; Lập lại so sánh CR với ký tự trong chuỗi
CPIR    ; BC giảm 1, HL tăng 1, nếu BC=0 hoặc (HL)=A dừng
LD      A,0FFH       ; Vì BC chứa số lần đếm âm nên lấy bù 2
SUB     C
LD      (0200H),A
HALT

```