1. 存储系统设计综合题：

We want to build up an embedded computer-based device to do test of SRAM chips. We do test by comparing the data write into and the data read back from a SRAM slot. If the data equal to each other, we say the slot is OK. And if all the slots are OK, then the chip is OK.

The device is required to start test automatically when booting up. There is a piece of ROM chip installed in the system to hold the boot up leader program. The ROM is assigned with an address starting from E0000H. And there is a far type jump instruction loaded at address FFFF0H, which will jump to address C0000H. All that means, we have to prepare another piece of ROM chip to hold our control program, and assign it with an address starting from C0000H. The circuit schema can be found in Figure 4. And the chips used in the device can be found in Table 1.

When the testing process is proceeding, if there is a un-functional slot founded, the process will be terminated, and a red light will be turned on. If the test is over and the chip is OK, a green light will be turned on. We make use of two piece flip-flop-based chip to interface the red and green lights.

Solve the problems below to complete the circuit design, and then prepare the control program.

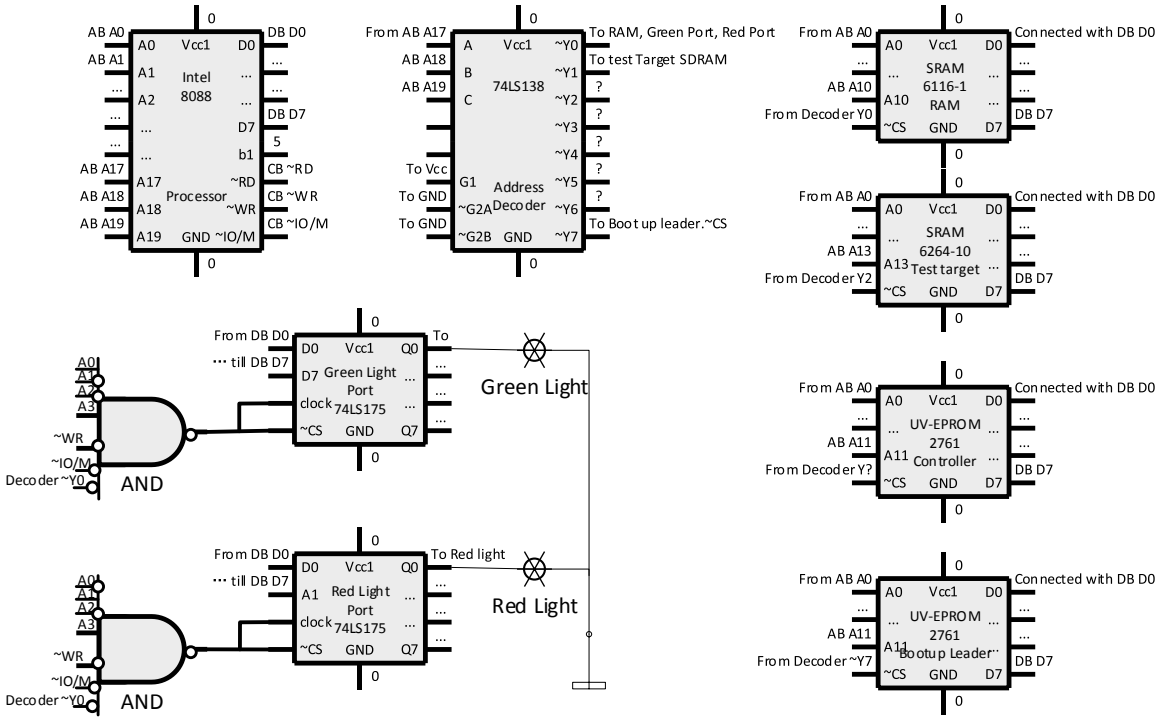



Figure 4 Circuit schema of SRAM testing device

Table 1 IC chips used in the device

| Chip ID | Type | Part Number | Capacity | Organization | Function |
|---|---|---|---|---|---|
| Processor | Intel 8088 | 80x88 | - | - | The Intel processor |
| Decoder | Address Decoder | 74LS138 | - | - | 3 to 8 address decoder |
| RAM | SRAM | 6116-1 | 16K | 2K*×8 | RAM space for the |

| | | | | | |
|---|---|---|---|---|---|
| | | | | | computer |
| Target | SRAM | 6264-10 | 64K | 8K**×8 | SRAM being tested |
| Controller | UV-EPROM | 2761 | 16K | 2K×8 | Hold the control program |
| BIOS | UV-EPROM | 2761 | 16K | 2K×8 | Boot up leader |
| Green Port | D-flipflop | 74LS175 | 8 D-flipflops | - | Implement a WR port |
| Red Port | D-flipflop | 74LS175 | 8 D-flipflops | - | Implement a WR port |

*2K = 2048 = $2^{11}$ =800H

**8K = 8192 = $2^{13}$ =2000H

Questions:

i) Witch one of the output pins of the 74LS138 should be taken as the chip selection signal of your controller program ROM? (score 3);

As our control program ROM must be installed in address C0000H, and C0000H is 1100 0000 0000 0000 0000,(A19A18A17=110), which means CBA = 110 =6H. So Y6 should be used as the ROM chip selection.

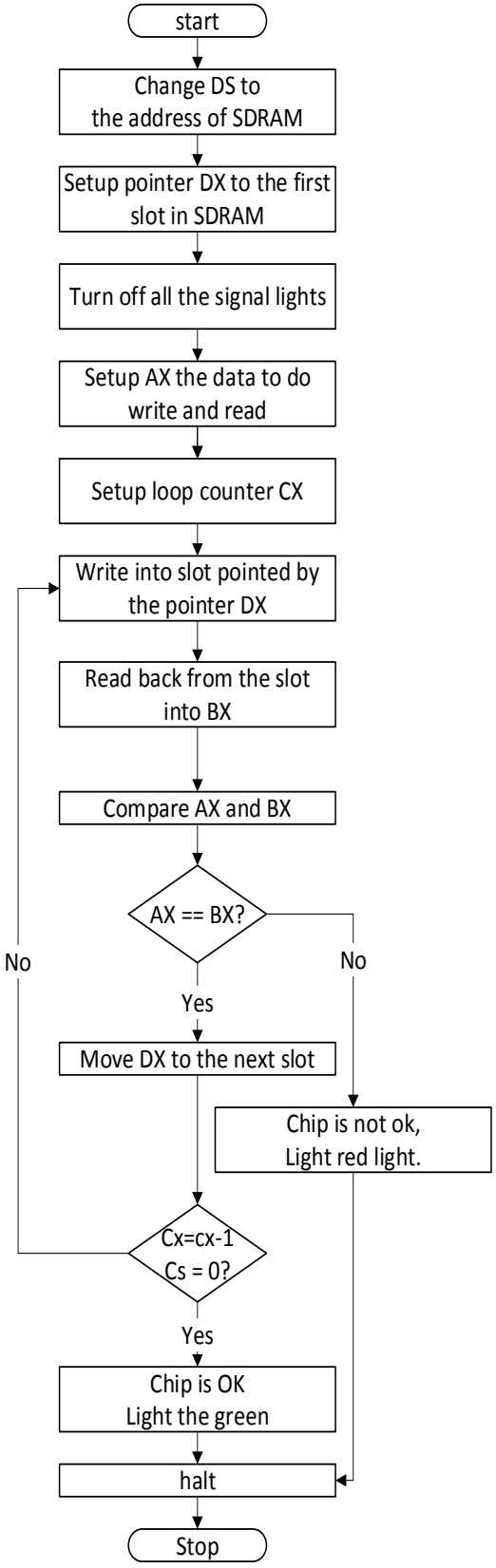
ii) Calculate the physical address assigned to the SRAM chip being tested. (score 3)

The SRAM being tested is actived by Y1.
So the address assigned to it ranges from 20000H to 21FFFH

iii) What is the port address assigned to the green light? And what is the port address assigned to the red light? (score 4)

Green actived by: 0000 1001B, so the port addr is 09H
Red actived by: 0000 1010B, so the port addr is 0AH

iv) Implement the control program. You can prepare the program follow the lead of the flow chart below, or you can show your own design. (score 10)




试题3 控制程序流程图

```
        .MODEL SMALL
        .STACK 64
        .DATA

        .CODE
MAIN    PROC  FAR
        ;MOV AX, @DATA
        ;MOV DS, AX
        ;TODO1: change DS to target SDRAM
        MOV AX, 2000H
        MOV DS, AX
        MOV DX, 00H
        ;TODO3:clear lights
        MOV AL, 00H
        OUT 09H, AL
        OUT 0AH, AL
        ;TODO4: begin loop
        MOV BL, 55H
        MOV CX, 2000H
LOOP_BEGIN:
        MOV [DX], BL
        MOV AL, [DX]
        CMP AL, BL
        JNZ LIGHT_RED
        ;TODO5: loop forward
        INC DX
        LOOP LOOP_BEGIN
LIGHT_GREEN:
        OUT 09H, 0FFH
        JMP HALT_PROCESSOR
LIGHT_RED:
        OUT 0AH, 0FFH;
HALT_PROCESSOR:
        HALT
        ;MOV AH, 4CH
        ;INT 21H            ;return to DOS
MAIN    ENDP
```

2. AD 采样及中断响应综合题

Please implements a voltage sampling system as described in the circuit schema figure K. Calculate the parameters of the interfacing circuit as required in the questions, and fill in the blanks within the control program to complete them. The designation of the circuit is described below:

i) The 8254.Timer0 is used to generate an interrupt every 0.5 second. The output of 8254.Timer0 is connected to 8259.IR0.

ii) The PIC controller 8259 is activated by 74LS138.Y1, single, edge triggering IRs, and IR0 is assigned “INT 50H”. The 8259 is in single buffered mode with normal EOI.

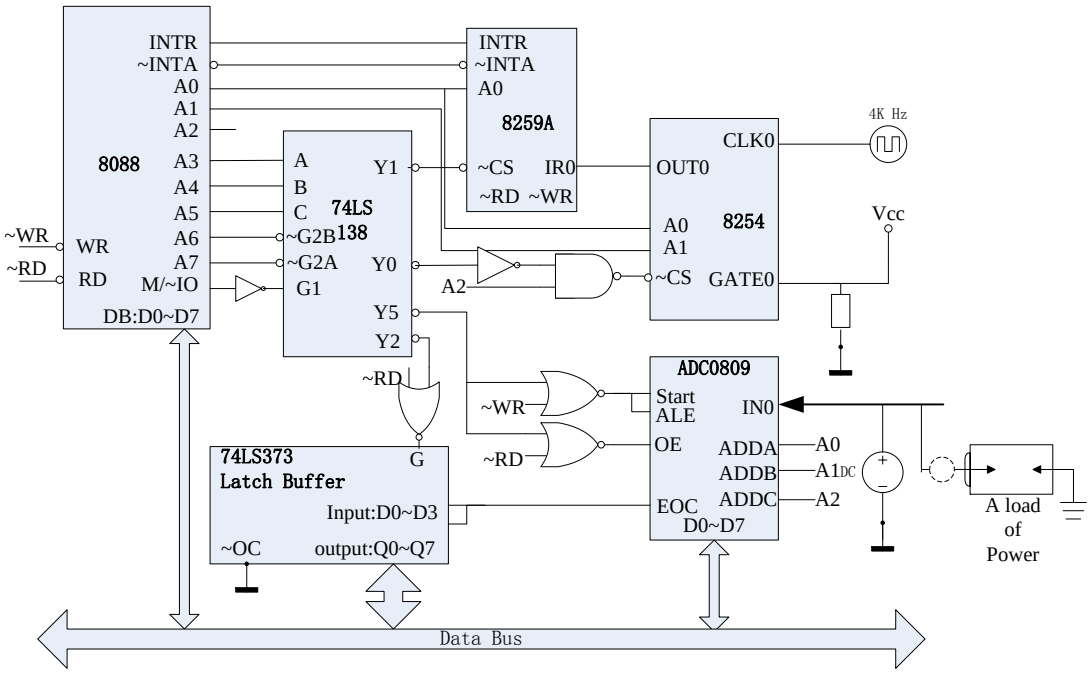
iii) In the interrupt service routine, we start a round of sampling process through channel IN0 of ADC0809, and read back the sample value. Then, we display the value into the center of the screen.

iv) We use the same interfacing circuit in experiment IV to connect the of ADC0809. Except that we use a direct input interface implemented with a piece of 74LS373. Through the direct input port, we can read back the state of ADC0809.EOC pin.

The problems are:

1) Calculate the port address for each chip, and fill in the table in the answer sheet.
2) Calculate the control word and initial value for 8254.Timer0, and fill in the blanks in the table in the answer sheet.
3) Calculate the control words of 8259 IR0, including the ICWs and OCWs, and fill in the blanks in the table below in the answer sheet.
4) Implement a control program to do initialization of 8254 and 8259, and register the interrupt service routine. Then, start to display the sample result.
5) Implement the ISR program to do sampling.

(Note: the register formats of all involved programmable controllers can be found in appendix)




| Question number | IC Chip and unit | Requirements | Answers | | | | notes |
|---|---|---|---|---|---|---|---|
| 1) | 8254 | Port addresses | Ranges from | 04H | to | 07H | |
| | 8259 | Port addresses | Ranges from | 08H | to | 09H | |
| | ADC0809 | Port addresses | Ranges from | 28H | to | 2FH | |
| | 74LS373 | Port address | 10H | | | | |
| 2) | 8254.Timer0 | Control word | 37H 0011 0111B | | | | |
| | | Initial value | 00H first, 20H next | | | | |
| 3) | 8259 | ICW1 | 13H, 0001 0011B | | | | |
| | | ICW2 | 50H, 0101 0000B | | | | |
| | | ICW3 | Do not needed | | | | |
| | | ICW4 | 09H, 0000 1001B | | | | |
| | | OCW1 | Do not needed | | | | |
| | | OCW2 | 20H | | | | |
| | | OCW3 | Do not needed | | | | |

**;Program for problem4 in final examination 2016-2017 autumn**
**;In this program, we use the ADC0809.CH0 to sample voltage.**
**;The 8 channels in ADC0809 has been mapped into 8 ports.**
**;Write the ADC.Port will start the sampling process;**
**;Read the ADC.Port will read back the sample value.**
**;The value read back is saved into variable SAMPLEVALUE**
**;and converted into ASCII code in variable DISPLAY_ASCII.**

```
;After all, the value is displayed in the center of the screen.
;THE PROGRAM UES KEYBOARD INPUT 'q' TO QUIT TO DOS
        .MODEL SMALL
        .STACK 64
        .DATA
        ;DEFINE    SYMBLE CONSTS
        PORT8254_Timer0    EQU 04H
        PORT8254_CtrlReg EQU 07H
        PORT8259_Low     EQU 08H
        PORT8259_High    EQU 09H
        PORTDIO_Input       EQU 10H
        PORTADC_CH0          EQU 28H
        CTRLWORD8254         EQU 36H
        CTRLWORD8259_ICW1 EQU 13H
        CTRLWORD8259_ICW2 EQU 50H
        CTRLWORD8259_ICW4 EQU 09H
        CTRLWORD8259_OCW2 EQU 20H
        ;DEFINE    VARIANTS
        SAMPLEVALUE DB 00H          ;sample value
        DISPLAY_ASCII DB 0,0,0,'$'      ;sample value in string format to be displayed
        .CODE
MAIN    PROC FAR        ;this is the program entry point
        MOV AX, @DATA ;load the data segment address
        MOV DS, AX ;assign value to data segment register
        ;TODO1: load the interrupt vector of our own ISR
;====solution1=========================================
        PUSH    DS              ;save current DS value
        MOV       AX, 0       ;set DS to the segment address of
        MOV       DS, AX ;DOS interrupt vector table
        MOV       BX, OFFSET myISR ;load IP address value of ISR
        MOV       SI, SEG myISR               ;load CS address value of ISR
        MOV       [0140H], BX      ;write IP into 0140H for INT 50H
        MOV       [0142H], SI       ;write CS into 0142H
        POP         DS              ;restore current DS value
;====solution2=========================================
        PUSH DS                                 ;save current DS value
        MOV AH, 25H                         ;DOS INT 21H function 25H
        MOV AL, 50H                         ;prepare interrupt number
        MOV DX, OFFSET MYISR     ;prepare IP of ISR into DX
        MOV BX, SEG MYISR               ;load CS of ISR
        MOV DS, BX                          ;prepare CS of ISR into DS
        INT 21H                             ;invoke the software interrupt
        POP DS                              ;restore current DS value
        ;TODO2: initialize 8259
```

```
        MOV     DX, PORT8259_Low
        MOV     AL, CTRLWORD8259_ICW1     ;ICW1
        OUT     DX, AL
        MOV     DX, PORT8259_High
        MOV     AL, CTRLWORD8259_ICW2     ;ICW2
        OUT     DX, AL
        MOV     AL, CTRLWORD8259_ICW4     ;ICW4
        OUT     DX, AL
        ;TODO3: Initialize 8254 and start timer0
        MOV DX, PORT8254_CtrlReg
        MOV AL, CTRLWORD8254
        OUT DX, AL
        MOV DX, PORT8254_Timer0
        MOV AL, 00
        OUT DX, AL
        MOV AL, 20
        OUT DX, AL              ;now the timer started
        ;TODO4: Clear screen and move cursor.
        MOV AX, 0600H
        MOV BH, 07H
        MOV CX, 0000
        MOV DX, 184FH
        INT 10H
        MOV AH, 02H
        MOV BH, 00H
        MOV DL, 38
        MOV DH, 12
        INT 10H
CONVERT_AND_DISPLAY:
        ;TODO5: Convert sample value into ASCII format
        MOV AL, BYTE PTR SAMPLEVALUE; load sample result
        MOV DI, OFFSET DISPLAY_ASCII;
        ADD DI, 2; move DI to the LSB digit
        MOV CX, 3
LOOPDIV:
        SUB AH, AH
        MOV BH, 10
        DIV BH          ; dinominator is the power=10D
        OR AH, 30H      ; convert BCD into ASCII
        MOV [DI], AH    ; keep remain as    lower digit
        DEC DI
        DEC CX
        CMP AL, BH      ;whether the quient is greater then the power
        JC LOOPDIV_OUT
```

```
        JMP LOOPDIV
LOOPDIV_OUT:
        OR AL, 30H
        MOV [DI], AL
        DEC DI
        DEC CX
        JZ DISPLAY_RESULT
        ;fill other digits with 0
CLRBUF:
        MOV BYTE PTR [DI], 20H   ;clear DISPLAY_BCD buffer with 'space'
        DEC DI
        LOOP CLRBUF
DISPLAY_RESULT:
        ;TODO6: Display the sample value with INT 21H
        MOV DX, OFFSET DISPLAY_ASCII
        MOV AH, 09H
        INT 21H
        ;TODO7: Test keyboard touch and quit
        MOV AH, 06H
        MOV DL, 0FFH
        INT 21H
        CMP AL, 'q'
        JE QUITDOS
        ;TODO8: delay and loop back to display new sample value
        CALL DELAY;
        JMP CONVERT_AND_DISPLAY;
        ;TODO9: quit to DOS
QUITDOS:
        MOV AH, 4CH     ;setup to
        INT 21H         ;return to DOS
MAIN        ENDP

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;Subprocedure: Delay
;;Function: DELAY for some milliseconds
DELAY   PROC NEAR
          PUSH BX
          PUSH CX
        MOV BX,0FFH
DELAY1: MOV CX, 0DFH
        LOOP $
        DEC BX
        JNZ DELAY1
        POP CX
```

```
        POP BX
        RET
DELAY   ENDP
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;Interrupt Service Routine:
MYISR   PROC FAR
        ;(the ISR program can be found in question 5) )
MYISR   ENDP

        END MAIN    ;this is the program exit point
```

5) The ISR program(score 4):

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;Interrupt Service Routine:
;;Function: start ADC and read back sample value to DS:SAMPLEVALUE
MYISR   PROC FAR
        PUSH DS
        PUSH DX
        PUSH AX
        MOV AX, @data
        MOV DS, AX
        STI
        ;TODO1: START SAMPLE
DOSAMPLE:
        MOV AL, 00H      ;Output a dummy value and start the ADC process
        MOV DX, PORTADC_CH0
        OUT DX, AL
        ;TODO2: TEST ADC.EOC state
TESTEOC:
        MOV DX, PORTDIO_Input
        IN AL, DX
        MOV AH, AL
        AND AH, 03H      ;mask off additional bits
        CMP AH, 03H
        JZ   TESTEOC
        ;TODO3: READ SAMPLE RESULT and save it
        MOV DX, PORTADC_CH0
        IN AL, DX
        MOV DX, OFFSET SAMPLEVALUE;
        MOV [DX], AL            ;save sample value in variable SAMPLEVALUE
        ;TODO4: Terminate Interrupt Handling Process and return
        MOV DX, PORT8259_High
        MOV AL, CTRLWORD8259_OCW2
        OUT DX, AL
```

```
        POP AX
        POP DX
        POP DS
        IRET
MYISR   ENDP
```