

Department of Electrical Engineering

University of Arkansas



ELEG3923 Microprocessor Final Review

Dr. Jingxian Wu

wuj@uark.edu

CH.5 ADDRESSING MODE

- **Bit addresses**

- RAM: 20H – 2FH bit addressable
 - Bit address range: 00H – 7FH
 - Given byte address and bit #, find bit address, and vice versa
- How do we tell from bit address from byte address?
 - Bit addressable instructions
 - SETB, CLR, CPL, JB, JNB, JBC, MOV bit, C
- Bit addressable SFRs
 - A, B, PSW, IP, IE, SCON, TCON, P1, P2, P3, P4
 - PCON is not bit addressable
- Bit directive

- **Extra 128-byte on-chip RAM**

- First 128 byte RAM: 00H – 7FH
- SFR address range: 80H – FFH
- upper 128-byte address range: 80H – FFH
- How to distinguish upper memory and SFR?
 - SFR: use name or direct addressing mode MOV 90H, #55H (MOV P1, #55H)
 - Upper memory: use indirect mode: MOV R0, 90H, MOV @R0, #55H

Byte address	General-purpose RAM							
7F								
30								
2F	7F	7E	7D	7C	7B	7A	79	78
2E	77	76	75	74	73	72	71	70
2D	6F	6E	6D	6C	6B	6A	69	68
2C	67	66	65	64	63	62	61	60
2B	5F	5E	5D	5C	5B	5A	59	58
2A	57	56	55	54	53	52	51	50
29	4F	4E	4D	4C	4B	4A	49	48
28	47	46	45	44	43	42	41	40
27	3F	3E	3D	3C	3B	3A	39	38
26	37	36	35	34	33	32	31	30
25	2F	2E	2D	2C	2B	2A	29	28
24	27	26	25	24	23	22	21	20
23	1F	1E	1D	1C	1B	1A	19	18
22	17	16	15	14	13	12	11	10
21	0F	0E	0D	0C	0B	0A	09	08
20	07	06	05	04	03	02	01	00
1F	Bank 3							
18								
17	Bank 2							
10								
0F	Bank 1							
08								
07	Default register bank for R0 - R7							
00								

CH.6 ARITHMETIC AND LOGICS

- **Arithmetic**

- Unpacked BCD, packed BCD
 - Addition of packed BCD; instruction: DA
- ADDC (addition with carry), e.g. 32B7H + B23AH
- SUBB (subtraction with borrow), e.g. 56A3H – 235CH
 - If result is negative, CY = 1, A stores 2's complement of absolute value.
- MUL AB, DIV AB (operands and results stored in registers A and B)
- OV flag is set to 1 if denominator is 0

- **Signed number operation**

- Signed 8-bit number (-128 – 127)
 - Find the 2's complement representation of decimal number and vice versa
- OV flag. Find the OV flag after ADD.
 - E.g. MOV A, #-128, ADD A, #-2

CH.6 ARITHMETIC AND LOGICS

- **Logic instructions**

- ANL, ORL, XRL
- CJNE dest, src, target
 - if dest \geq src, CY = 0
 - If dest $<$ src, CY = 1

- **Rotation and serialization**

- RR, RL, RRC, RLC
- SWAP

- **BCD and ASCII**

- Packed BCD \rightarrow unpacked BCD \rightarrow ASCII
- Binary (decimal) \rightarrow ASCII
 - Keep dividing by 10

CH.7 PROGRAMMING IN C

- **Data type**

- unsigned char, signed char, unsigned int, signed int
- sbit (bit addressable sfr), bit (bit addressable RAM), sfr
- sbit mybit = 10 ; the address of MYBIT is 10
- bit mybit = 10 ; the value of mybit is 10
- Time delay: it's difficult to determine the exact delay of loops in C

- **logic operators**

- Byte logic operation: && (and), || (or), ! (not)
- Bit logic operation: & (and), | (or), ^ (xor), ~ (not), << (shift left), >> (shift right) (**not cyclic shift**)

- **Data conversion**

- Packed BCD → unpacked BCD → ASCII
- Binary (or decimal) → ASCII

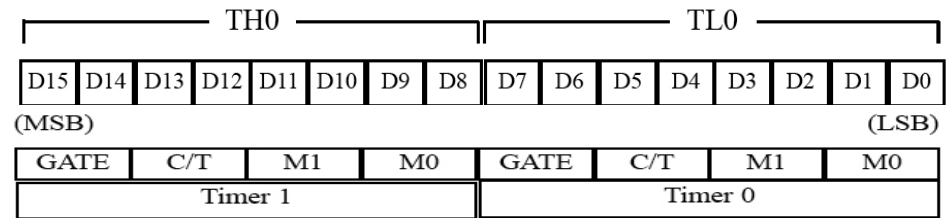
- **RAM and ROM**

- Declare variable in ROM:
 - code unsigned char mystring[] = "abcde"

CH.9 TIMER

- **Registers**

- TH0, TL0
- TMOD



- **Timer**

- Procedure:
 - 1. initialize TMOD; 2. load initial value; 3. start timer;
 - 4. monitor TF_n or use interrupt; 5. when timer expires, stop timer; 6 clear flag
- Modes
 - Mode 1 (16-bit), mode 0 (13-bit), mode 2 (8-bit auto-reload)
- Initial value v.s. timer delay
 - 16-bit timer: $(65535 - \text{nnnn} + 1) * \text{clock period} = \text{time delay}$
 - 8-bit timer: $(255 - \text{nn} + 1) * \text{clock period} = \text{time delay}$
 - |negative initial value| = # of clock ticks
 - Frequency of clock ticks: XTAL/12

CH.9 TIMER

- **Counter (C/T = 1)**
 - C/T = 0: timer, clock source is XTAL/12
 - C/T = 1: counter, clock source is external pulses connected through P3.4 (T0) or P3.5 (T1)
 - Counting events happening outside of 8051
 - Procedure: 1. init TMOD; 2. clear THn and/or TLn; 3. set P3.4 or P3.5 as input; 4. start counter (timer)
 - Operation modes are the same as timer

CH.10 SERIAL PORT

- **Basics of serial communication**

- Serial v.s. parallel; Simplex, half-duplex, full-duplex
- Asynchronous communication, framing
- Transfer rate
 - Baud (symbol rate), bit rate
 - Terminology: storage 1Kilo = 2^{10} , data rate 1Kilo = 1,000
- RS232
 - DTE, DCE, straight through cable, null model cable
 - 3-wire RS232, TXD, RXD, GND
 - Signal level: '1': -3V to -25V, '0': 3V to 25V
 - MAX232: voltage converter
 - UART
 - Baud rate
 - Relationship between #of ticks and baud rate
- SCON register

SM0	SM1	SM2	REN	TB8	RB8	TI	RI
-----	-----	-----	-----	-----	-----	----	----

CH.10 SERIAL PORT

- **Tx**

- 1. Initialize timer and baud; 2. initialize SCON; 3. start timer; 4 MOV SBUF, A; 5. monitor TI (or use interrupt); 6. clear TI

- **Rx**

- 1. Initialize timer and baud; 2. initialize SCON; 3. start timer; 4. monitor RI (or use interrupt); 5. MOV A, SBUF; 6. clear RI

- **PCON (not bit addressable)**

- double baud

SMOD	--	--	--	GF1	GF0	PD	IDL
------	----	----	----	-----	-----	----	-----

- **2nd serial port**

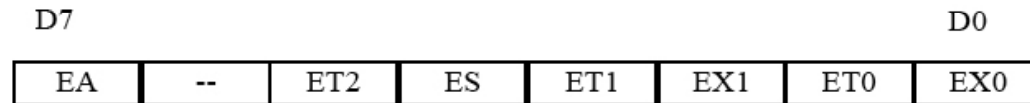
- Use different SCON and SBUF registers
- All remaining registers are the same as the 1st serial port

CH.11 INTERRUPT

- **Interrupt**

- Six interrupts: timer (2), serial port (1), external (2), reset (1)
- ISR, interrupt vector table
- Registers

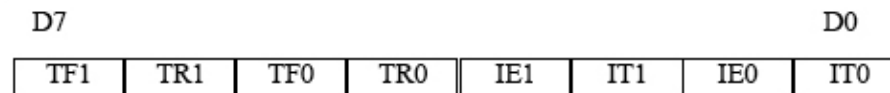
- IE:



- IP:



- TCON:



- **Timer interrupt**

- Triggered by TF_n.
- TF_n will be cleared automatically by RETI

- **External interrupt**

- INT0 (EXT0), INT1 (EXT1)
- Level triggered, edge triggered (IT_n of TCON)
- IEn: edge triggered interrupt flag
 - IEn is cleared automatically by RETI

CH.11 INTERRUPT

- **Serial port interrupt**
 - TI and RI share the same interrupt and ISR
 - We must check TI and RI flag in ISR
 - TI and RI need to be cleared manually in ISR
 - Interrupt for TI and polling for RI
- **Interrupt priority**
 - Default priority
 - Set priority with IP register
 - Trigger the interrupt by software

CH.14 MEMORY

- **Terminologies**

- Primary memory; secondary memory
- Capacity; organization; speed
- PROM, EPROM, EEPROM, Flash
- SRAM, DRAM, SDRAM

- **Interfacing**

- Data bus, control bus, address bus (calculation of address range)

- **External code ROM**

- EA, PSEN, ALE, P0 (data, lower byte of address), P2 (higher byte of addr)

- **External data ROM and RAM**

- RD v.s. PSEN
- MOVX
- Interfacing
- On-chip SRAM

- **C programming**

- XBYTE