Embedded Controller Programming

Counters, Timers and I/O in Assembly Language

Ken Arnold

 1

Outline

- Timer/Counters
- Serial Port
- More 8051 Instructions
- Examples

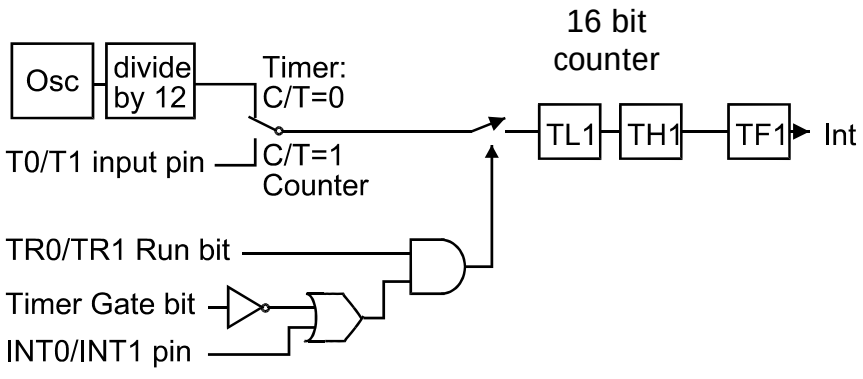
 2

Timer/Counters

- Control Register
- Modes: (Timers 0,1)
  - Counting, Timing
  - One shot 16 bit (mode 1)
  - Auto reload 8 bit (mode 2)
  - Timer 2 Enhanced modes:
    - 16 bit auto reload
    - Capture/compare

 3

8051 Timer Mode 1



 4

Timer Control TCON 088h

7	6	5	4	3	2	1	0
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

- TFx Timer x Overflow Flag (input)
- TRx Timer x Run Enable (control)
- IEx Ext Int Detect Flag (input)
- ITx Ext Int Edge Sensitive 1=Edge 0=Level

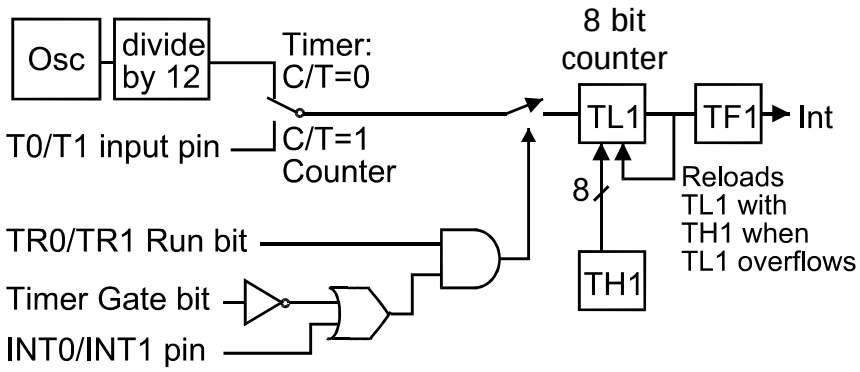
 5

Timer Flags

- Timers set flag on overflow
  - Count increments every 12 clock cycles
- Mode 1 used for a single 16 bit delay
  - One-shot delay, then it must be reloaded
- Mode 2 reloads automatically, 8 bit delay
  - TLx reloaded with THx when TLx overflows
  - Results in constant interrupt frequency
- Load timer with *2's complement* of count!

 6

8051 Timer Mode 2



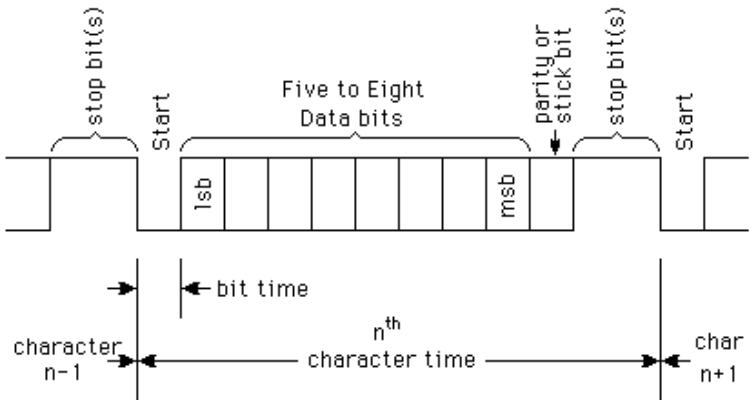
 7

Serial Data

- Synchronous (clocked):
  - Separate data and clock signals
  - Shift register serial mode 0
  - Also I²C™, SPI™, Microwire™, One-Wire™
- Asynchronous (unclocked like COM port):
  - Data signal only, serial modes1-3
  - Independent clock in receiver
  - UART: Universal Async Receiver Transmitter

 8

Async Serial Data Format

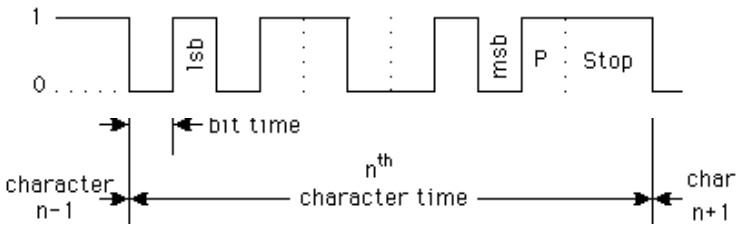


 9

Serial Data Example

Serial data 4DH: ASCII code for upper case ‘M’

Binary 0100 1101 - LSB is sent first



 10

Serial Port Modes

- Serial Modes:
  - 0: Ext shift register data on RxD, Clk on TxD
  - 1: 8 bit UART (like the PC's serial COM port)
  - 2: 9 bit, UART fixed clock freq X/64 or X/32
  - 3: 9 bit, UART variable clock frequency
- Mode 1 most commonly used for RS-232
- Timer 1 Used for Data Bit Rate, Mode 2

 11

Serial Control SCON 098h

7	6	5	4	3	2	1	0
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

- RI=1 when Serial Receive Buffer is Full
- TI=1 when Serial Transmit Buffer is Empty
- MUST be cleared using CLR TI or CLR RI

 12

SCON cont'd

7	6	5	4	3	2	1	0
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

- SM0, SM1 Select Modes 0..3
- SM2=1: Multiprocessor RI set if RB8=1
- REN=1 Enables Receiver
- RB8, TB8 Ninth bit received or transmitted

 13

Assembler

- Overview of asm51.pdf Manual
- Syntax
- Notation
- Pseudo-Ops
- Directives

 14

SDK Monitor Commands

- Review Monitor Commands
- ? Or Help Command: lists commands
- Reserved Names / SFRs
- Assemble/Disassemble
- PS: Through/over Calls
- Go vs. Execute

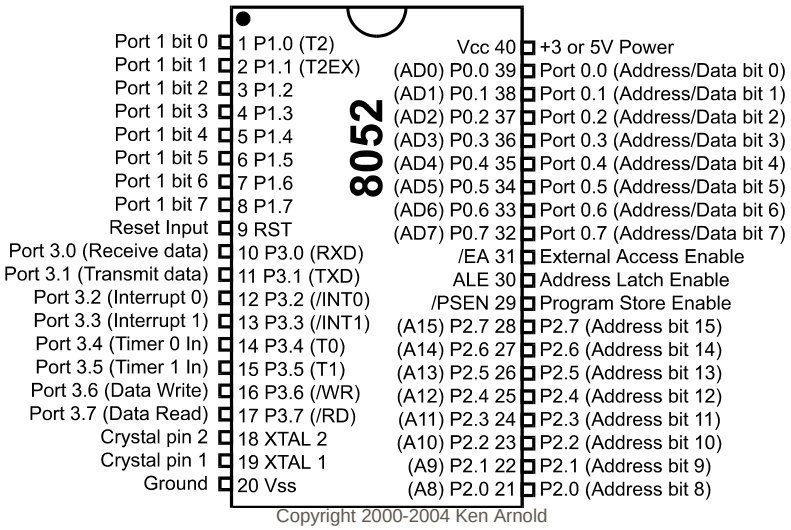
 15

Example Programs

- Simple P1 I/O
- Counters:
  - counter1.asm, counter2.asm
- Timers:
  - timer0.asm, timer1.asm, timer2.asm
- Serial I/O:
  - serial1.asm, serial2.asm, serial3.asm

 16

8052 Pin Definitions



 17

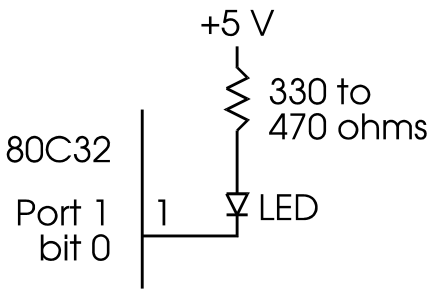
Port I/O *for the 8051 family*

- Ports are SFRs in Internal Data Memory
- Port is 0ffh (same as 0xff in C) after Reset
- Port 1 is uncommitted - location 90h
- Bit addressable from bits 90h to 97h
  - LSB address = 90h, MSB address = 97h
- SDK Port 1 on DIP plug pins 1..8
  - Access with monitor command P1

 18

Driving an LED directly

- 8051 Port I_{OL} >> I_{OH}
- Drive LED from 5V
- LED OFF when bit= 1
- LED ON when bit= 0
- For LED with Vf=~2V
  - voltage across R is: +5V - 2V = 3V
  - current in R and LED: 3V/330 ohms = 9 mA



 19

8051 Programming Model

- General Architecture
  - Harvard - Separate Program and Data Memory
  - Accumulator Based
- Memory Spaces and Data Paths
  - Multilevel Data Access: Byte and Bit Level
- Instruction Set
  - CISC - Complex Instruction Set

 20

Programmer's CPU Model

- Address Mode Notation
- Types of Operands
- Memory Addresses
- Access Methods

 21

Address Notation

- Rn range: R0..R7
  - Register Rn of the current Register Bank.
- direct range: 00-FFh
  - 8-bit internal data location's address.
  - Internal Data RAM location (0-7Fh) or
  - SFR (80-FFh) SFRs are usually referenced by symbolic name, e.g. P1 rather than 90h.

 22

Address Notation 2

- @Ri range: @R0 or @R1 only
  - 8-bit internal data location (0-FFh) addressed indirectly through register R1 or R0.
- #data range: 0-FFh
  - 8-bit constant included in the instruction
- #data 16 range: 0-FFFFh
  - 16-bit constant included in the instruction

 23

Address Notation 3

- addr 16 range: 0-FFFFh
  - 16-bit destination address. Used by LCALL and LJMP. A branch can be anywhere within the 64k-byte Program Memory address space.
- addr 11 range: 0-7FFh
  - 11-bit destination address. Used by ACALL and AJMP. The branch will be within the same 2k-byte page of program memory as the first byte of the following instruction.

 24

Address Notation 4

- rel (RELative) Range: -128 to +127
  - Signed (two's complement) 8-bit offset byte. Used by SJMP and all conditional jumps
  - Range is –128 to +127 bytes relative to first byte of the following instruction
- bit Range: 0-FFh
  - Bit in Internal Data RAM (bit address 0-7Fh)
  - Special Function Register bits (address 80-FFh)

 25

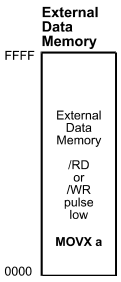
Data Pointer

- dptr: 16 bit address pointer
  - Initialize with MOV dptr,#data16
  - Upper and lower bytes are also SFRs:
    - High and low byte of dptr are dph and dpl
- Used for references to Address Spaces:
  - External Data
    - MOVX a,@dptr or MOVX @dptr,a
  - Code Memory
    - MOVC a,@a+dptr note address is @[a+dptr]

 26

External Data Access

- External Data - Indirect Only
  - Address range 0000-FFFFh
  - Using 16 Bit Data Pointer (dptr)
    - MOVX a,@dptr ; load a from xdata
    - MOVX @dptr,a ; store a in xdata
  - Address range 00-FFh using 8 bit pointers @R0/@R1
    - MOVX a,@R0 ; indirect from xdata
    - MOVX a,@R1 ; range 0-FFh
    - MOVX @R0, a
    - MOVX @R1, a



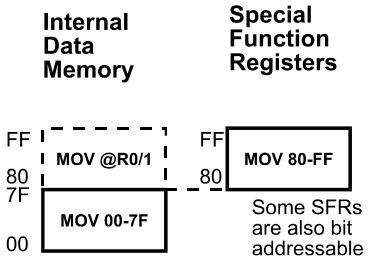
Using DPTR (Data Pointer)

```
        mov   dptr,#uartsr    ; Address of uart status reg
        movx  a,@dptr         ; Get status byte from uart

        etc...                ; ...code continues
                              ; Memory mapped I/O
uartsr  equ   0fff9h          ; UART status register
uartd   equ   0fffbh          ; UART data register
rxrdy   equ   01h             ;  receive buffer full
txrdy   equ   04h             ;  transmit buffer empty
```


Data Memory Access

- Internal Data
- Range 00-7Fh
  - Direct address, 8 bits: 0-7Fh
  - MOV a,nnh where nn<80h
  - MOV a,@R0 where R0<80h
- Range 80-FFh
  - Indirect Access ONLY
  - Addresses 80-FFh
    - MOV a,@Ri
  - Examples:
    - MOV a,@R0
    - MOV direct, @R0

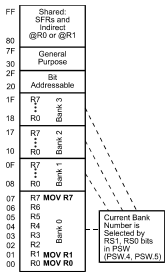


- Direct addr >=80h is SFR
  - Example
    - MOV a,90h ; get Port1

 29

Internal Data Memory

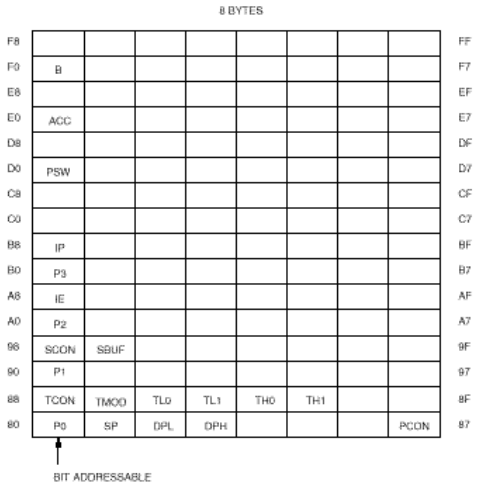
- Registers R0..R7
  - 8 registers per bank
  - 4 Banks available
- Bit Addressable 20-2Fh
- General Purpose 30-7Fh
- SFRs:
  - Accumulator, I/O
  - PSW, B, Counter/Timer
  - Misc. Registers



 30

SFR Addresses

- SFRs 80-FFh
- Some are Bit Addressable (if last 3 bits of SFR address =0)
  - ex:P1.3 is bit 93h
- Available SFRs Depend on CPU variant used



 31

Control Transfer

- JMP addr
  - Go to address
- JMP @a+dptr
  - Indirect Jump
  - Address @(a+dptr)
- JZ rel
  - if Accumulator is Zero
- JNZ rel
  - if Acc. is Non-Zero
- CALL addr
  - Save PC on stack
  - Go to subroutine address
- RET return
  - Pop PC from stack
- RETI
  - Return from ISR
  - Allows Interrupts

 32

Control Transfer 2

- CJNE Compare and jump if Not Equal
  - CJNE @Ri,#data,rel
  - CJNE a,#data,rel
  - CJNE a,direct,rel
  - CJNE Rn,#data,rel
- DJNZ Decrement and Jump if Not Zero
  - Loop counter: DJNZ counter,address
  - DJNZ direct,rel
  - DJNZ Rn,rel

 33

Data Path: Logic

- CLR A
  - clears A
- CPL A
  - one's complement of A
- Rotate Accumulator:
  - RR A lsb -> msb
  - RL A msb -> lsb
  - RRC A lsb -> c -> msb
  - RLC A msb -> c -> lsb
- Bit-wise Logical
  - Destination is A:
    - ANL A, x
    - ORL A, x
    - XRL A, x
    - x Operands can be:
    - #data, Rn, @Ri, direct
  - Destination is Direct
    - ANL direct, #data
    - ANL direct, a

 34

Bit Operations

- MOV (c is carry bit)
  - MOV c,bit
  - MOV bit,c
- CLR clear bit
  - CLR c CLR bit
- SETB set bit
  - SETB c SETB bit
- CPL complement bit
  - CPL c CPL bit
- ANL
  - Logical AND
  - ANL c,bit ; c AND bit
  - ANL c,/bit ; c AND NOT bit
- ORL
  - Logical OR
  - ORL c,bit ; c OR bit
  - ORL c,/bit ; c OR NOT bit

 35

Bit Conditional Jumps

- JB jump if bit = 1
  - JB bit,rel
- JNB jump if bit = 0
  - JNB bit,rel
- JC jump if carry = 1
  - JC rel
- JNC jump if carry = 0
  - JNC rel
- JBC jump on bit & clr
  - JBC bit,rel
  - jump if bit = 1, clear bit
  - Can be used for semaphore operations
  - Indivisible test and clear
  - a.k.a. mutex

 36

Conditional Branching

```
low:                    ; waits until external
      jb    P1.0, low   ; switch pulls P1.0 low
      call  delay       ; debounce delay
      ...
high:                   ; waits until external
      jnb   P1.0, high  ; switch is opened
      ....
```

 37

Summary

- Timer/Counters
- Serial Port
- More 8051 Instructions
- Examples

 38