8051 Microcontroller

EE4380 Fall 2001

Pari vallal Kannan
Center for Integrated Circuits and Systems
University of Texas at Dallas

8051 Architecture

- Programmer's View
  - Register Set
  - Instruction Set
  - Memory map
- Designer's View
  - Pinout
  - Timing characteristics
  - Current / Voltage requirements

Programmer's View – Register Set

- Registers
  - A, B, R0 to R7 : 8 bit registers
  - DPTR : [DPH:DPL] 16 bit register
  - PC : Program Counter (Instruction Ptr) 16bits
  - 4 sets of R0-R7
  - Stack pointer SP
  - PSW Program Status Word (Flags)
    - Carry CY, Aux Carry AC, Reg Bank selector, Overflow, Parity
  - Special Function Registers (SFRs)
    - Ports, Timers, Interrupt (enable, priority), Serial port, power

Assembly – Absolute Basics

- Intel Assembly format

  Operation destination source ; comment

- Values are to be preceded by a # sign
  - #55, #32 etc
- Hex values are to be followed by H
  - #55H, #32H
- If the first figure in a hex quantity is a letter (A-F) then a 0 ***must*** precede it
  - #0FFH, #0C1H, #0D2H
- No operation : NOP !

Register Set – Accumulator A, ACC

- Commonly used for *mov* and arithmetic
- Implicitly used in opcodes or use ACC or 0E0H
- Example of Implicit reference
  - Instruction : mov A, R0
  - Opcode : E8
  - The Accumulator is implicitly coded in the opcode
- Explicit reference to Accumulator
  - Instruction : push ACC
  - Opcode: C0 E0

Register Set – B Register

- Commonly used as a temporary register, much like a 9th R register
- Used by two opcodes
  - mul AB, div AB
- B register holds the second operand and will hold part of the result
  - Upper 8bits of the multiplication result
  - Remainder in case of division

Register Set – R0 to R7

- Set of 8 registers R0, R1, … R7, each 8 bit wide
- Widely used as temporary registers
- Available in 4 banks (effectively 4x8 registers)
- Bank is chosen by setting RS1:RS0 bits in PSW
- Default bank (at power up) is the bank0

Registers - DPTR

- 16 bit register, called Data Pointer
- Used by commands that access external memory
- Also used for storing 16bit values

```
mov DPTR, #data16      ; setup DPTR with 16bit ext address
movx A, @DPTR          ; copy mem[DPTR] to A
```

- DPTR is useful for string operations, look up table (LUT) operations

Registers - PC

- PC is the program counter
- Referred to as the Instruction Pointer (IP) in other microprocessors
- PC points to the next program instruction *always*
- After fetching an instruction (1 or multi byte), PC is automatically incremented to point to the next instruction
- Cannot directly manipulate PC (exceptions JMP statements)
- Cannot directly read contents of PC (tricks available)

Registers - SP

- SP is the stack pointer
- SP points to the last used location of the stack
  - Push operation will first increment SP and then copy data
  - Pop operation will first copy data and then decrement SP
- In 8051, stack grows upwards (from low mem to high mem) and can be in the internal RAM only
- On power-up, SP is at 07H
- Register banks 2,3,4 (08H to 1FH) is the default stack area
- Stack can be relocated by setting SP to the upper memory area in 30H to 7FH
  - mov SP, #32H

Registers - PSW

- Program Status Word is a bit addressable 8bit register that has all the flags
- CY Carry Flag – Set whenever there is a carry in an arithmetic operation
- AC Aux. Carry Flag – Carry from D3 to D4. Used for BCD operation
- P Parity Flag – P=1 if A has odd number of 1s
  - Even parity
- OV Overflow Flag – Set if any operation causes an overflow

Flags - Illustration

- Addition example

	38	0011 1000
+	2F	0010 1111
	---	---
	67	0110 0111
	---	---

CY = 0

AC = 1

P = 1

Registers - SFRs

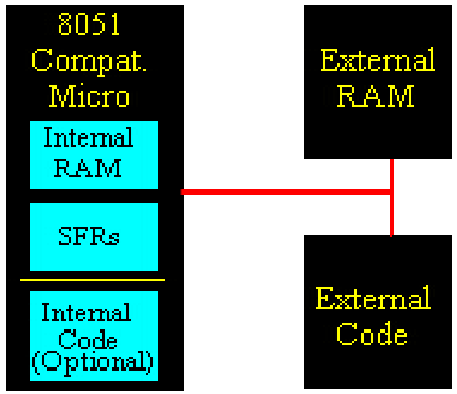
- Special Function Registers at direct addresses 80H to FFH

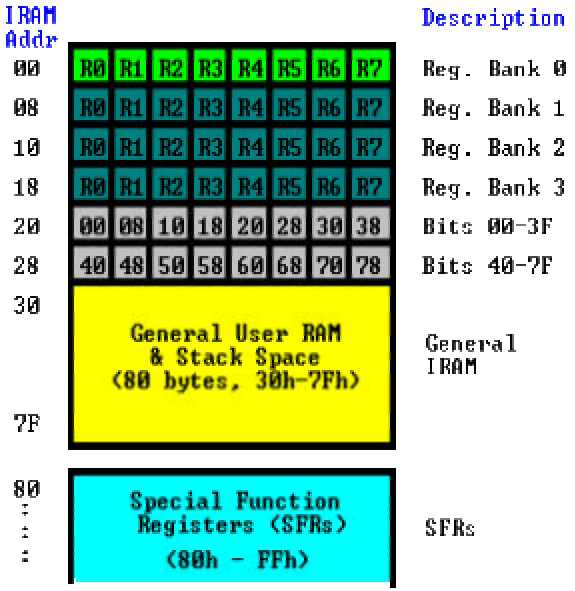
80	P0	SP	DPL	DPH				PCON	87
88	TCON	TMOD	TL0	TL1	TH0	TH1			8F
90	P1								97
98	SCON	SBUF							9F
A0	P2								A7
A8	IE								AF
B0	P3								B7
B8	IP								B9
C0									C7
C8									CF
D0	PSW								D7
D8									DF
E0	ACC								E7
E8									EF
F0	B								F7
F8									FF

Blue background are I/O port SFRs
Yellow background are control SFRs
Green background are other SFRs

8051 - Memory Map

Memory Type	Start	End	Signal	Instruction
Internal RAM	00H	7FH		MOV A, xxH
External RAM	0000H	FFFFH	RD, WR	MOVX A, @DPTR
External ROM	0000H	FFFFH	PSEN	MOVC, MOVX
Internal ROM	0000H	????H		MOVC





- Internal ROM is vendor dependant
- On power-up PC starts at 0000H in ROM space

8051 – Instruction Set

- Data Transfer

  mov, movc, movx, push, pop, xch, xchd
- Logical

  anl, orl, xrl, clr, cpl, rl, rlc, rr, rrc, swap
- Arithmetic

  add, addc, subb, inc, dec, mul, div
- Program control

  jmp, ajmp, ljmp, sjmp, jc, jnc, jb, jnb, jbc, jz, jnz, acall, lcall, cjne, djnz, ret, reti
- NOP

8051 Assembly Introduction

- Assembly statement structure

  [label:] opcode [operands] [;comment]

- Example

  start: mov A, #D0H ;code starts here

- Assembler directives
  - ORG xxxxH : origin, start assembling at xxxxH
  - EQU : define a constant
    - count EQU 25
  - DB : define byte, defines data
    - DATA1: DB 28
    - DATA2: DB “hello world”
  - END : end of assembly file

Assembly Design Flow

- Create the assembly source file test.asm
- Assemble the asm file
  - as51 test.asm
  - Assembler produces error and code list in test.lst
  - If no errors, assembler produces .obj file
- Link the .obj files to produce an .abs file
- Create hex file from the .abs file
- Most assemblers directly produce the .hex file
- Download the .hex file onto the board or burn it into an eprom.