EE337: MICROCONTROLLERS AND MICROPROCESSORS

RAJBABU VELMURUGAN AND SHANKAR BALACHANDRAN

Embedded C

Using C

- Becoming very common for programming microcontrollers
- Different flavors possible with different tools
- Embedded C using Keil is very convenient and useful

Keywords

Keil C51 compiler adds few more keywords to the scope C Language:

_at_	far	sbit
alien	idata	sfr
bdata	interrupt	sfr16
bit	large	small
code	pdata	_task_
compact	_priority_	using
data	reentrant	xdata

Data/idata

```
unsigned char data x;
//or
unsigned char idata y;
```

Decides whether to store in internal data memory

bdata

unsigned char bdata x;

//each bit of the variable x can be accessed as follows

x ^ 1 = 1; //1st bit of variable x is set

x ^ 0 = 0; //0th bit of variable x is cleared

_at_

- unsigned char idata x _at_ 0x30;
- // variable x will be stored at location 0x30
- // in internal data memory

sbit

- sbit Port0_0 = 0x80;
- // Special bit with name Port0_0 is defined at address 0x80

Defines a special bit from special function register (SFR)

Sfr

- sfr Port1 = 0x90;
- // Special function register with name Port1 defined at addrress 0x90

sfr16

- sfr16 DPTR = 0x82;
- // 16-bit special function register starting at 0x82
- // DPL at 0x82, DPH at 0x83

using

```
void function () using 2
{
// code
}
// Funtion named "function" uses register bank 2 while executing its code
```

Interrupt

- Tells the compiler that a function is an ISR
- 32 interrupts are possible

Interrupt Number	Address
0	0003h
1	000Bh
2	0013h
3	001Bh
4	0023h
5	002Bh
6	0033h
7	003Bh
8	0043h
9	004Bh
10	0053h
11	005Bh
12	0063h
13	006Bh
14	0073h
15	007Bh

Interrupt Number	Address
16	0083h
17	008Bh
18	0093h
19	009Bh
20	00A3h
21	00ABh
22	00B3h
23	00BBh
24	00C3h
25	00CBh
26	00D3h
27	00DBh
28	00E3h
29	00EBh
30	00F3h
31	00FBh

Example

```
void External_Int0() interrupt 0
{
//code
}
```

Memory Models

- Small
  - All variables in internal data memory.
- Compact
  - Variables in one page, maximum 256 variables (limited due to addressing scheme, memory accessed indirectly using r0 and r1 registers);
- Large
  - All variables in external ram. variables are accessed using DPTR.

Usage

- //For Small Memory model
- #pragma small
- //For Compact memory model
- #pragma compact
- //For large memory model
- #pragma large

Pointers

- Can do all operations that C has
- Two types
  - Generic – just like in C
  - Memory specific

Memory Specific

- Can specify pointers to be of specific type
- char data *c;
  - //Pointer to character stored in Data memory
- char xdata *c1;
  - //Pointer to character stored in External Data Memory.
- char code *c2;
  - //Pointer to character stored in Code memory

Functions

- [Return_type] Fucntion_name ([Arguments]) [Memory_model] [reentrant] [interrupt n] [using n]
- Return_type
  - The type of value returned from the function. If return type of a function is not specified, int is assumed by default.
- Function_name
  - Name of function
- Arguments
  - Arguments passed to function

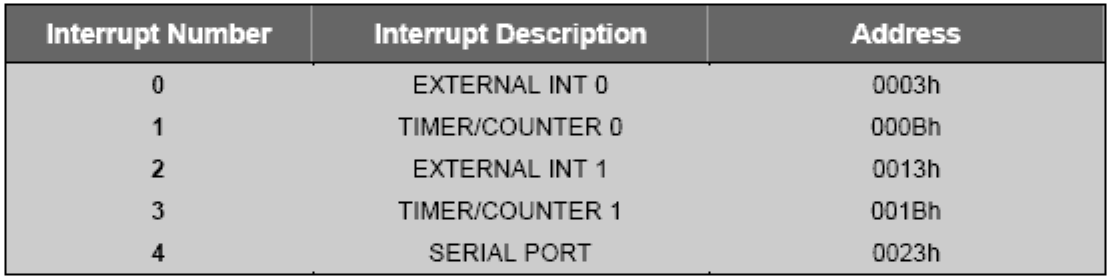
Other options for functions

- Example: int add_number (int a, int b) Large
- reentrant
  - To indicate if the function is reentrant or recursive.
- interrupt
  - Indicates that function is an interrupt service routine.
- using
  - Specify register bank to be used during function execution.

Example

```
void function_name () using 2
{
//function uses Bank 2
//function code
}
```

Default INterrupts



Interrupt Number	Interrupt Description	Address
0	EXTERNAL INT 0	0003h
1	TIMER/COUNTER 0	000Bh
2	EXTERNAL INT 1	0013h
3	TIMER/COUNTER 1	001Bh
4	SERIAL PORT	0023h

ISR Example

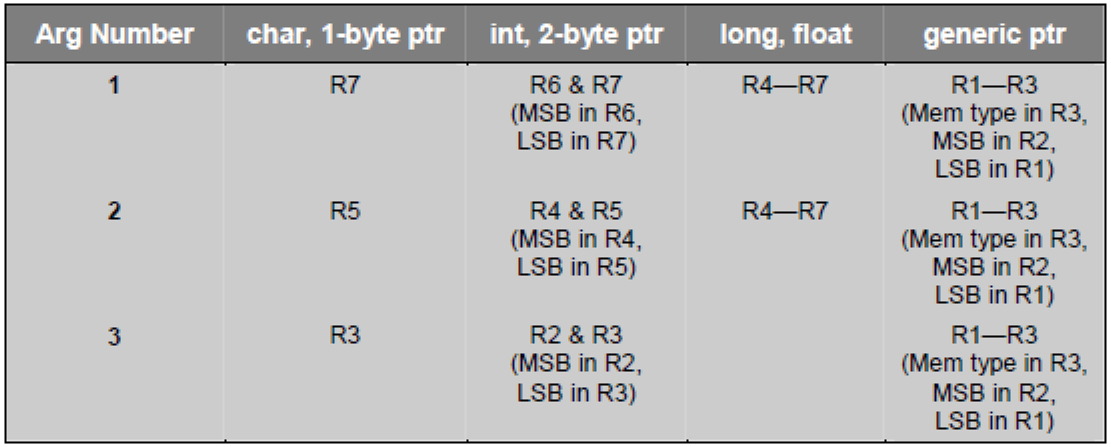
```
void isr_name (void) interrupt 2
{
// Interrupt routine code
}
```

Function parameters

- Can be passed using registers

Or

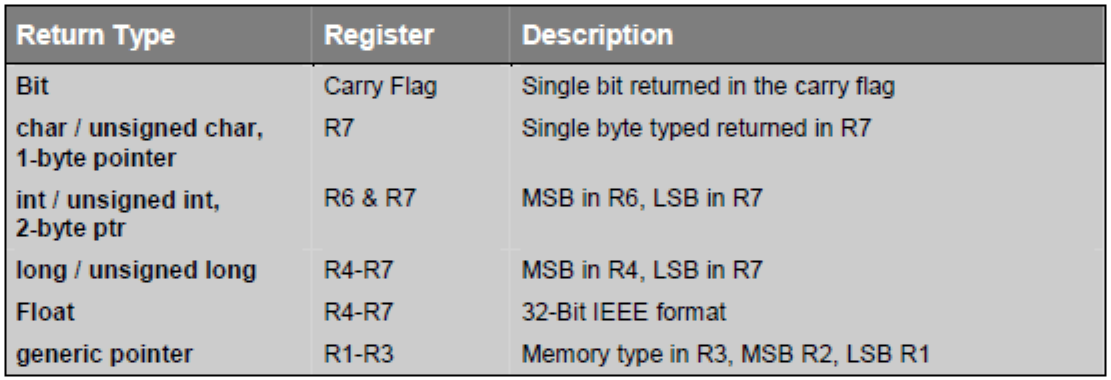
- Fixed memory locations



Arg Number	char, 1-byte ptr	int, 2-byte ptr	long, float	generic ptr
1	R7	R6 & R7 (MSB in R6, LSB in R7)	R4—R7	R1—R3 (Mem type in R3, MSB in R2, LSB in R1)
2	R5	R4 & R5 (MSB in R4, LSB in R5)	R4—R7	R1—R3 (Mem type in R3, MSB in R2, LSB in R1)
3	R3	R2 & R3 (MSB in R2, LSB in R3)		R1—R3 (Mem type in R3, MSB in R2, LSB in R1)

Return

- Always in registers



Return Type	Register	Description
Bit	Carry Flag	Single bit returned in the carry flag
char / unsigned char, 1-byte pointer	R7	Single byte typed returned in R7
int / unsigned int, 2-byte ptr	R6 & R7	MSB in R6, LSB in R7
long / unsigned long	R4-R7	MSB in R4, LSB in R7
Float	R4-R7	32-Bit IEEE format
generic pointer	R1-R3	Memory type in R3, MSB R2, LSB R1

Note

- Use C for large programs
- Can also use C And assembly together