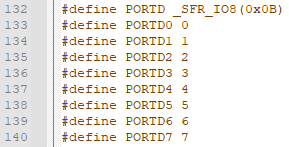
查看手册《Atmel-7810-Automotive-Microcontrollers-ATmega328P\_Datasheet.pdf》page280页：



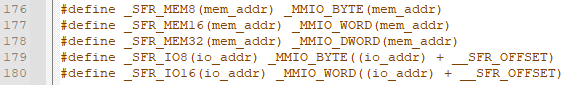
可知PORTD寄存器的相对地址是0x0B，绝对地址是0x2B。二者相差0x20，即32，表示32个通用寄存器R0~R32所占据的地址：



在iom328p.h文件中查看PORTD的定义：



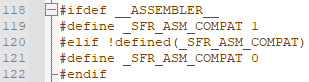
而\_SFR\_IO8()在sfr\_defs.h文件中定义：



\_MMIO\_BYTE()也在sfr\_defs.h文件中定义：

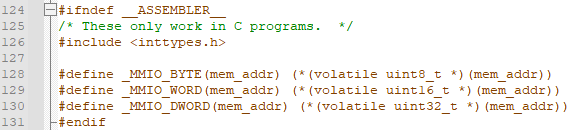


下面分析一下sfr\_defs.h文件：



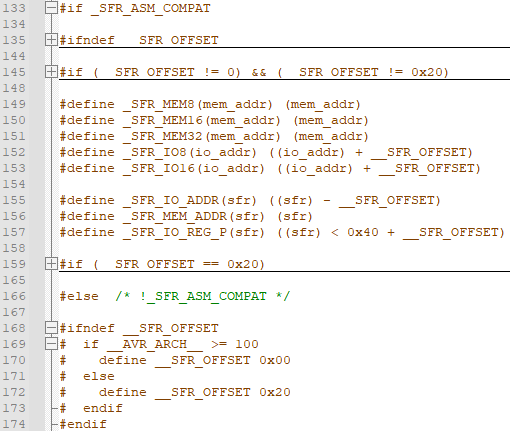
从118行开始，如果定义了\_\_ASSEMBLER\_\_，则定义\_SFR\_ASM\_COMPAT为1，否则如果没有定义\_SFR\_ASM\_COMPAT，则定义\_SFR\_ASM\_COMPAT为0。解析：如果在Makefile中选择的是汇编器，则编译器内部会定义\_\_ASSEMBLER\_\_。但现在Makefile中选择的是C语言编译器avr-gcc.exe，则\_\_ASSEMBLER\_\_并没有被定义。故118~122的预处理语句执行完毕后，得到\_SFR\_ASM\_COMPAT被定义为0。

接着看124~131行：



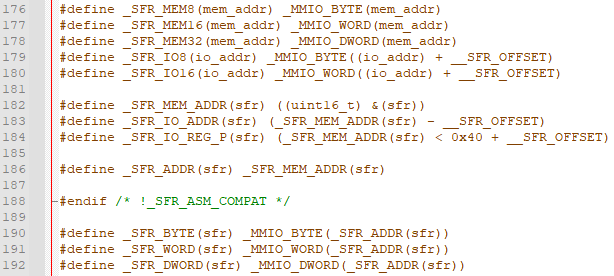
由于没有定义\_\_ASSEMBLER\_\_，故会执行126~130行的代码。需要注意的是\_MMIO\_BYTE()，例如\_MMIO\_BYTE(0x2B)，宏展开后为(\*(volatile uint8\_t \*)(0x2B))，它表示先是用(volatile uint8\_t \*)(0x2B)将0x2B强制类型转换为指针，即地址，然后再\*(volatile uint8\_t \*)(0x2B)，即表示地址为0x2B的单元格中的内容。由于前面已知PORTD的地址为0x2B，故(\*(volatile uint8\_t \*)(0x2B))就表示PORTD寄存器的内容。

接着看133~174行：

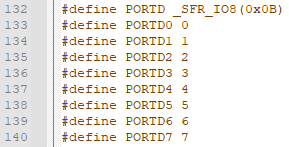


由于前面定义\_SFR\_ASM\_COMPAT为0，故135~165行不再执行，而执行168~172行。由于还没有定义\_\_SFR\_OFFSET，故执行168~172内部的#if语句。注意\_\_AVR\_ARCH\_\_，它是芯片型号决定的，由于在Makefile中选择的是atmega328p，从文档《avr8-gnu-toolchain-3.7.0.1796-readme.pdf》查得atmega328p的架构是AVR5，故编译器会定义\_\_AVR\_ARCH\_\_为5。这样，进而执行172行，定义\_\_SFR\_OFFSET为0x20，即64个IO寄存器地址相对于32个通用寄存器R0~R31地址的偏移量0x20。

接着看看176~192行：



其中要注意179行的\_SFR\_IO8()，从该宏定义看见其接受的是IO寄存器的相对地址。回到iom328p.h文件看看PORTD的定义：

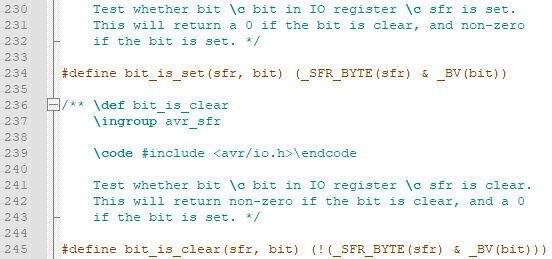


这样，PORTD就定义为地址为0x2B的SRAM单元格中的内容，即PORTD寄存器的值。

接着回到sfr\_defs.h，有几个有用的宏定义：

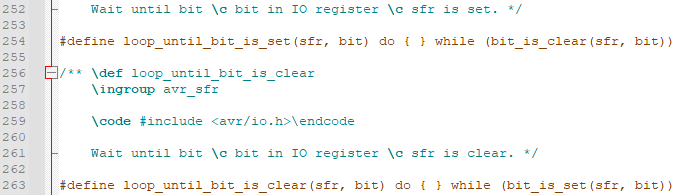


它表示1左移bit位。



bit\_is\_set()用于测试sfr寄存器的某个位是否为1，若该位为1，则bit\_is\_set()返回非0数值；若该位为0，则bit\_is\_set()返回0。例如bit\_is\_set(DDRB,5)表示判断DDRB寄存器的位5是否为1。

bit\_is\_clear()用于测试sfr寄存器的某个位是否为0，若该位为0，则bit\_is\_clear ()返回非0数值；若该位为1，则bit\_is\_clear ()返回0。例如bit\_is\_clear (DDRB,5)表示判断DDRB寄存器的位5是否为0。



loop\_until\_bit\_is\_set()用于等待sfr寄存器的某个位为1，若该位为0，则一直等待，直到该位为1，则结束等待。

loop\_until\_bit\_is\_clear()用于等待sfr寄存器的某个位为0，若该位为1，则一直等待，直到该位为0，则结束等待。