APPENDIX F

AVR FUSE BITS

OBJECTIVES

Upon completion of this appendix, you will be able to:

>> Explain the role of the fuse bytes in an AVR-based system
>> Explain the role of brown-out detection voltage (BOD) in system reset

SECTION F.1: ATMEGA328 FUSE BITS

There are some features of the AVR that we can choose by programming the bits of fuse bytes.

ATmega328 has three fuse bytes. Tables F-1 through F-3 give a short description of the fuse bytes. Notice that the default values can be different from production to production and time to time. In this section we examine some of the basic fuse bits. The Atmel website (http://www.atmel.com) provides the complete description of fuse bits for the AVR microcontrollers. It must be noted that if a fuse bit is incorrectly programmed, it can cause the system to fail. An example of this is changing the SPIEN bit to 1, which disables SPI programming mode. In this case you will not be able to program the chip any more! Also notice that the fuse bits are '0' if they are programmed (enabled) and '1' when they are not programmed.

In addition to the fuse bytes in the AVR, there are 6 lock bits to restrict

Table F-1: Extended Fuse Byte for ATmega328

Extended Fuse Byte	Bit No.	Description	Default Value
BODLEVEL	2	Brown-out detector trigger level	1 (unprogrammed)
BODLEVEL	1	Brown-out detector trigger level	1 (unprogrammed)
BODLEVEL	0	Brown-out detector trigger level	1 (unprogrammed)

Table F-2: Fuse Byte (High)

High Fuse Byte	Bit No.	Description	Default Value
RSTDISBL	7	Exernal Reset Disable	1 (unprogrammed)
DWEN	6	debugWIRE Enable	1 (unprogrammed)
SPIEN	5	Enable SPI serial program and data downloading	0 (programmed)
WDTON	4	Watchdog Timer Always On	1 (unprogrammed)
EESAVE	3	EEPROM memory is preserved through the chip erase	1 (unprogrammed)
BOOTSZ1	2	Select boot size	0 (programmed)
BOOTSZ0	1	Select boot size	0 (programmed)
BOOTRST	0	Select reset vector	1 (unprogrammed)

Table F-3: Fuse Byte (Low)

Low Fuse Byte	Bit No.	Description	Default Value
CKDIV8	7	Divide clock by 8	0 (programmed)
CKOUT	6	Clock output	1 (unprogrammed)
SUT1	5	Select start-up time	1 (unprogrammed)
SUT0	4	Select start-up time	0 (programmed)
CKSEL3	3	Select clock source	0 (programmed)
CKSEL2	2	Select clock source	0 (programmed)
CKSEL1	1	Select clock source	1 (unprogrammed)
CKSEL0	0	Select clock source	0 (programmed)

access to the Flash memory. These allow you to protect your code from being copied by others. In the development process it is not recommended to program lock bits because you may decide to read or verify the contents of Flash memory. Lock bits are set when the final product is ready to be delivered to market. In this book we do not discuss lock bits. To study more about lock bits you can read the data sheets for your chip at http://www.atmel.com.

Fuse bits and oscillator clock source

As you see in Figure F-1, there are different clock sources in AVR. You can choose one by setting or clearing any of the bits CKSEL0 to CKSEL3.

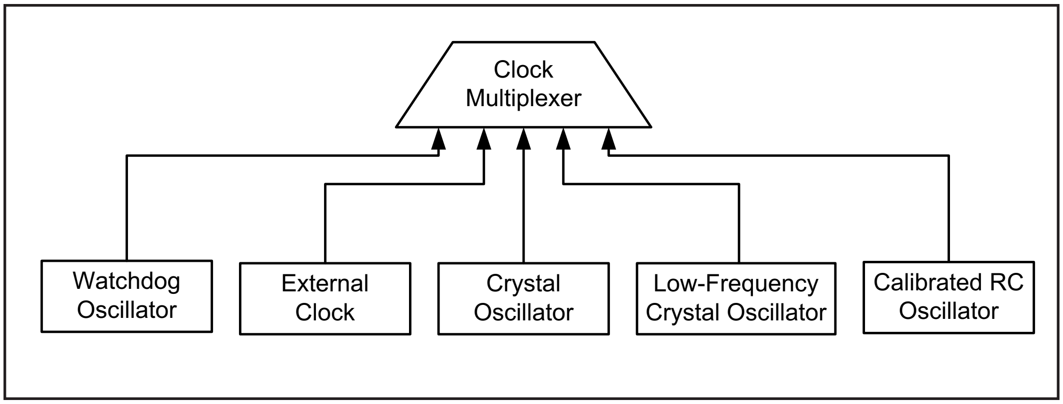

Figure F-1. ATmega328 Clock Sources

Table F-4: Internal RC Oscillator Operation Modes

CKSEL3...0	Clock Source
0000	External Clock
0010	Calibrated Internal 8MHz RC Oscillator
0011	Internal 128kHz RC Oscillator
0101- 0100	Low Frequency Crystal Oscillator
0111- 0110	Full Swing Crystal Oscillator
1111- 1000	Low Power Crystal Oscillator

CKSEL0–CKSEL3

The four bits of CKSEL3, CKSEL2, CKSEL1, and CKSEL0 are used to select the clock source to the CPU. The default choice is internal 8MHz RC (0010), which uses the on-chip RC oscillator. In this option there is no need to connect an external crystal and capacitors to the chip. It must be noted that using an internal RC oscillator can cause about 10% inaccuracy and is not recommended in applications that need precise timing.

By default, the CKDIV8 (Divide clock by 8) is also programmed. So, the default frequency is 1MHz.

By setting CKSEL0...3 bits to 0000, we can use an external clock source for the CPU. In Figure F-2 you see the connection to an external clock source.

The most widely used option is to connect the XTAL1 and XTAL2 pins to a crystal (or ceramic) oscillator, as shown in Figure F-3. In this mode, when the CKSEL bits are set to 0111-0110, the oscillator output will oscillate with a full rail-to-rail swing on the output, causing a more powerful clock signal. This is suitable when the chip drives a second clock buffer or operates in a very noisy environment. As you see in Table F-5, this mode has a wide frequency range. When the CKSEL bits are set to 1000-1111, the oscillator has a smaller output swing and a limited frequency range. This mode cannot be used to drive other clock buffers, but it does reduce power consumption considerably. There are four choices for the crystal oscillator option. Table F-5 shows all of these choices. Notice that mode 1100 cannot be used with crystals, and only ceramic resonators can be used. Example F-1 shows the relation between crystal frequency and instruction cycle time.

In Arduino trainer boards, a 16MHz crystal is connected to the XTAL pins and the CKSEL bits are set to 1111 (Low Power Crystal Oscillator).

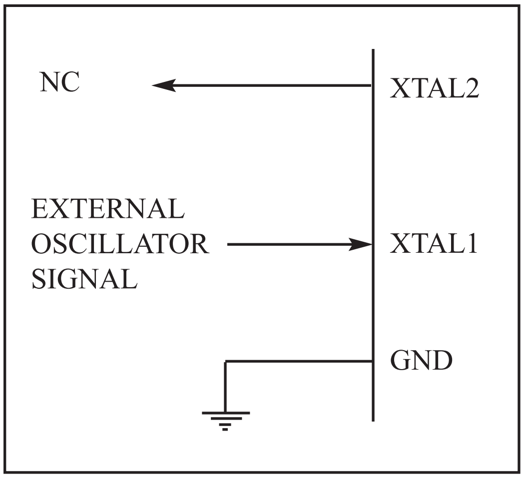

Figure F-2. XTAL1 Connection to an External Clock Source

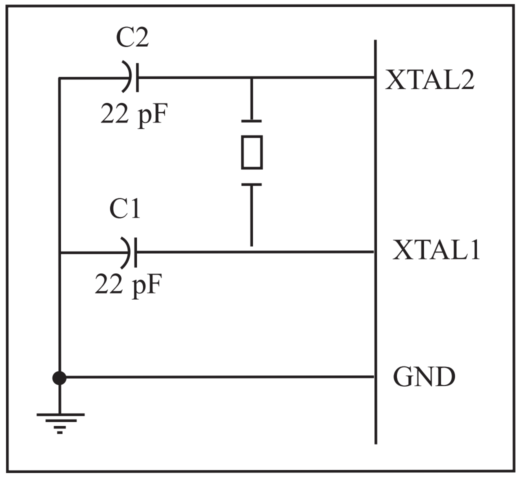

Figure F-3. XTAL1–XTAL2 Connection to Crystal Oscillator

Table F-5: ATmega328 Crystal Oscillator Frequency Choices and Capacitor Range

CKSEL3...1	Frequency (MHz)	C1 and C2 (pF)	
0110	0.4–20	12–22	Full Swing
1100	0.4–0.9	Not for crystals	Low Power
1101	0.9–3.0	12–22	Low Power
1110	3.0–8.0	12–22	Low Power
1111	8.0–16.0	12–22	Low Power

Example F-1

Find the instruction cycle time for the ATmega328 chip with the following crystal oscillators connected to the XTAL1 and XTAL2 pins. CKDIV8 is not programmed.

(a) 4 MHz (b) 8 MHz (c) 10 MHz

Solution:

(a) Instruction cycle time is 1/(4 MHz) = 250 ns
(b) Instruction cycle time is 1/(8 MHz) = 125 ns
(c) Instruction cycle time is 1/(10 MHz) = 100 ns

Fuse bits and reset delay

The most difficult time for a system is during power-up. The CPU needs both a stable clock source and a stable voltage level to function properly. In AVRs, after all reset sources have gone inactive, a delay counter is activated to make the reset longer. This short delay allows the power to become stable before normal operation starts. You can choose the delay time through the SUT1, SUT0, and CKSEL0 fuses. Table F-6 shows start-up times for the different values of SUT1, SUT0, and CKSEL fuse bits and also the recommended usage of each combination. Notice that the third column of Table F-6 shows start-up time from power-down mode. Power-down mode is not discussed in this book.

Table F-6: Startup Time for Crystal Oscillator and Recommended Usage

CKSEL0	SUT1...0	Start-Up Time from Power-Down	Delay from Reset (VCC = 5)	Recommended Usage
0	00	258 CK	4.1ms	Ceramic resonator, fast rising power
0	01	258 CK	65ms	Ceramic resonator, slowly rising power
0	10	1K CK	-	Ceramic resonator, BOD enabled
0	11	1K CK	4.1ms	Ceramic resonator, fast rising power
1	00	1K CK	65 ms	Ceramic resonator, slowly rising power
1	01	16K CK	-	Crystal oscillator, BOD enabled
1	10	16K CK	4.1 ms	Crystal oscillator, fast rising power
1	11	16K CK	65 ms	Crystal oscillator, slowly rising power

Brown-out detector

Occasionally, the power source provided to the V_{CC} pin fluctuates, causing the CPU to malfunction. The ATmega family has a provision for this, called *brown-out detection*. The BOD circuit compares VCC with BOD-Level and resets the chip if VCC falls below the BOD-Level. The BOD-Level can be selected using the BODLEVEL fuses as shown in Table F-7. When VCC increases above the trigger level, the BOD circuit releases the reset, and the MCU starts working after the time-out period has expired.

Table F-7: BODLEVEL Fuse Coding

BODLEVEL2...0	BOD-Level
000-011	Reserved
100	4.3 V
101	2.7 V
110	1.8 V
111	BOD Disabled

Review Questions

1. A given ATmega328-based system has a crystal frequency of 16 MHz. What is the instruction cycle time for the CPU?
2. How many fuse bytes are available in ATmega328?
3. True or false. Upon power-up, both voltage and frequency are stable instantly.
4. True or false. The internal RC oscilator works for the frequency of 8MHz.
5. Which fuse bits are used to enable the BOD?
6. True or false. Upon power-up, the CPU starts working immediately.
7. The brown-out detection voltage can be set at _______ or ______ by________ fuse bits.
8. True or false. The higher the clock frequency for the system, the lower the power dissipation.

SUMMARY

In this section, the fuse bytes were discussed. We use fuse bytes to enable features such as BOD and clock source and frequency.

PROBLEMS

SECTION F.1: AVR FUSE BITS

1. Which clock sources does the AVR have?
2. What fuse bits are used to select clock source?
3. Which clock source do you suggest if you need a variable clock source?
4. Which clock source do you suggest if you need to build a system with minimum external hardware?
5. Which clock source do you suggest if you need a precise clock source?
6. How many fuse bytes are there in the AVR?
7. Which fuse bit is used to set the brown-out detection voltage for the ATmega328?

20. Which fuse bits are used to enable the brown-out detection voltage for the ATmega328?
21. If the brown-out detection voltage is set to 4.0 V, what does it mean to the system?

ANSWERS TO REVIEW QUESTIONS

SECTION F.1: AVR FUSE BITS

1. 1/16 MHz = 62.5 ns
2. 3 bytes
3. False
4. True
5. BODLEVEL
6. False
7. 1.8 V, 2.7 V, 4.3 V, BODLEVEL
8. False