DAFTAR PUSTAKA

Atmel. *Microcontroller, All Data Sheet*, 2010

Cahttophadyay, D, *Dasar Elektronika*, penerjemahSutanto UI Press, Jakarta Indonesia

Fairchild, *Data Sheet IC Regulator*, 2010

Iswanto,. *Design dan Inplementasi Nsistem Embeded Mikrokontroler Atmega8535 Dengan Bahasa Basic*. Gava Media.

Wardhana, L. (2006). *Mikrokontroler AVR Seri Atmega8535 Simulasi, Hardware, dan Aplikasi*. Yogyakarta: Penerbit Andi.

Winoto, A. (2010). *Mikrokontroler AVR ATmega 8/ 32/ 16/ 8535 dan Pemrogramannya dengan Bahasa C pada WinAVR*, Bandung

Widodo Budiharto, (2011). *Aneka Proyek Mikrokontroler*, Graha Ilmu, Yogyakarta

Wiryadinata, R., Th. S. Widodo, W. Widada, Sunarno, (2008), *High-Pass Filter Effect on Error Corection for Inertial Navigation System Algorithm*, SIPTEKGAN XII.

Pitowarno, E., *Teknik Desain Robot*. Penerbit Andi : Yogyakarta, 2006.

Zuhal., (2004), *Prinsip Dasar Elektronika*. Jakarta: Gramedia Pustaka Utama

Vishay. *LCD, All Data Sheet*, 2010

Malviano, Albert P., *Prinsip-prinsip Elektronika*, terjemahan oleh Hanapi Gunawan, Erlangga Jakarta

Milman dan Halkiss : *Integrated Electronics, Analog and Digital, Circuit and System*, terjemahan oleh M. Barmawi dan M.O Tjia, Erlangga, Jakarta

Metzger, Daniell I,. *Electronics Component, Instrument, And Troubleshooting*, Printice Hall Inc.

Schultz, Mithehel E., *Electronics Devices, A Text and Software Problem manual*, McGraw Hill Publishing Company Ltd, New Delhi

LAMPIRAN

http://teknikelektronika.com , diakses 24 April 2016

http://belajar-dasar-pemrograman.blogspot.sg/2013/04/pengenalan-atmega8535.html, diakses pada tanggal 24 April 2016

http://argi-argianto.blogspot.co.id/2011/02/atmega-8535.html, diakses 28 April 2016

http://belajar-dasar-pemrograman.blogspot.sg/2013/04/pengenalan-atmega8535.html, diakses pada tanggal 28 April 2016

http://ilmuinstrumentasi.blogspot.co.id/2013/03/ldr-light-dependent-resistor.html, diakses pada tanggal 29 April 2016

http://ilmuinstrumentasi.blogspot.co.id/2013/03/ldr-light-dependent-resistor.html, diakses pada tanggal 29 April 2016

http://teknikelektronika.com/pengertian-piezoelectric-buzzer-cara-kerja-buzzer/, diakses pada tanggal 30 April 2016

Cooding Program

$regfile = "m8535.dat" → mikrokontroler yang digunakan 8535 AVR

$crystal = 8000000

Config Lcdpin = Pin , Db4 = Portb.4 , Db5 = Portb.5 , Db6 = Portb.6 , Db7 = Portb.7 , E = Portb.3 , Rs = Portb.2 → konfigurasi *pin LCD* ke mikrokontroler

Config Lcd = 16 * 2

Config Adc = *Single* , *Prescaler* = *Auto* , *Reference* = Avcc → konfigurasi *ADC*

Declare Sub Sensor_1(byval Ch0 As Word)
Declare Sub Sensor_2(byval Ch1 As Word)
Declare Sub Sensor_3(byval Ch2 As Word)
Declare Sub Sensor_4(byval Ch3 As Word)
Declare Sub Delay_1()

Config Portc.3 = Output →
Config Portc.0 = Output →
Config Portc.1 = Output →
Config Portc.2 = Output →

Error1 Alias Portc.3
Error2 Alias Portc.0
Error3 Alias Portc.1
Error4 Alias Portc.2

Dim Ldr_1 *As Single*→ deklarasi Ldr_1 bertipe *single*

Dim Ldr_2 *As Single*→ deklarasi Ldr_2 bertipe *single*

Dim Ldr_3 *As Single*→ deklarasi Ldr_3 bertipe *single*

Dim Ldr_4 *As Single*→ deklarasi Ldr_4 bertipe *single*

Dim Detik_1 *As Word*→ deklarasi Detik_1 bertipe *word*

Dim Detik *As Word*→ deklarasi Detik bertipe *word*

Dim Dataadc_0 *As Word*→ deklarasi Dataadc_0 bertipe *word*

Dim Dataadc_1 *As Word*→ deklarasi Dataadc_1 bertipe *word*

Dim Dataadc_2 *As Word*→ deklarasi Dataadc_2 bertipe *word*

Dim Dataadc_3 *As Word* → deklarasi Dataadc_3 bertipe *word*

Cursor Off → matikan cursor di *LCD*

Cls → *Clear Screen LCD*

Do → intruksi untuk menjalankan suatu perintah

One: → sub menu dengan nama "One"

Call Delay_1() → panggil waktu tundaan

Call Sensor_1(0) → panggil sub menu sensor 1

Call Sensor_2(1) → panggil sub menu sensor 2

If Detik > 7 *Then* : Detik = 0 → jika detik lebih besar dari 7 maka detik =0

Goto Two → pergi ke menu sub *Two*

Cls → *Clear Screen LCD*

End If

Loop → perulangan suatu perintah

Do → intruksi untuk menjalankan suatu perintah

Two: → sub menu dengan nama "two"

Call Delay_1()() → panggil waktu tundaan

Call Sensor_3(2) → panggil sub menu sensor 1

Call Sensor_4(3) → panggil sub menu sensor 2

If Detik > 7 *Then* : Detik = 0→ jika detik lebih besar dari 7 maka detik =0

Goto One →pergi ke menu *sub one*

Cls → *Clear Screen LCD*

End If

Loop → perulangan suatu perint

Sub Sensor_1(ch0 *As Word*)
Start Adc
Dataadc_0 = Getadc(0) → ambil data dari *ADC* 0 mikrokontroller
Ldr_1 = Dataadc_0
Ldr_1 = Ldr_1
Locate 1 , 8
Lcd "" ; Fusing(ldr_1 , "#.#")
Lcd " "
Locate 1 , 1
Lcd "LDR1 = "
If Ldr_1 > 100 *Then*
Error1 = 1
Elseif Ldr_1 < 99 *Then*
Error1 = 0
End If
End Sub

Sub Sensor_2(ch1 As Word)
Start Adc
Dataadc_1 = Getadc(1)
Ldr_2 = Dataadc_1
Ldr_2 = Ldr_2
Locate 2 , 8
Lcd "" ; Fusing(ldr_2 , "#.#")
Lcd " "
Locate 2 , 1
Lcd "LDR2 = "
If Ldr_2 > 100 Then
Eror2 = 1
Elseif Ldr_2 < 99 Then
Eror2 = 0
End If
End Sub

Sub Sensor_3(ch2 As Word)
Start Adc
Dataadc_2 = Getadc(2)

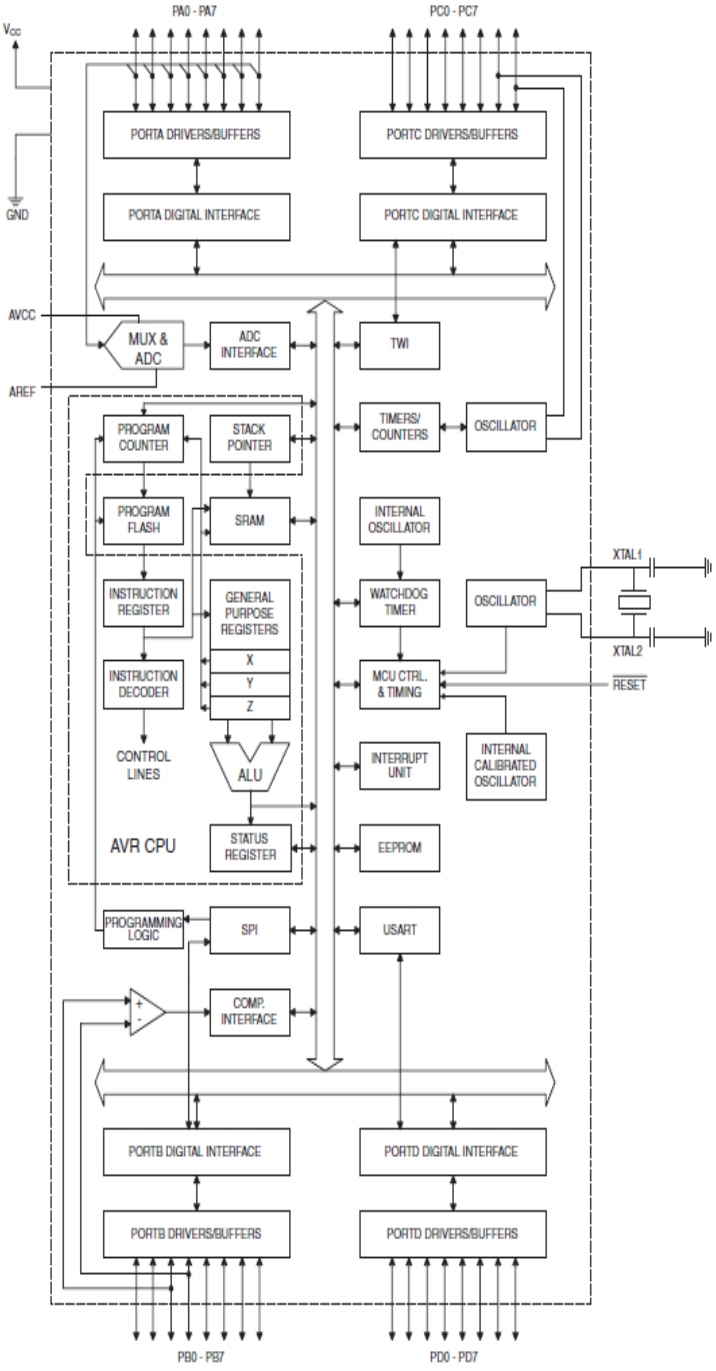
```
Ldr_3 = Dataadc_2
Ldr_3 = Ldr_3
Locate 1 , 8
Lcd "" ; Fusing(ldr_3 , "#.#")
Lcd "     "
Locate 1 , 1
Lcd "LDR3 = "
If Ldr_3 > 100 Then
Eror3 = 1
Elseif Ldr_3 < 99 Then
Eror3 = 0
End If
End Sub

Sub Sensor_4(ch3 As Word)
Start Adc
Dataadc_3 = Getadc(3)
Ldr_4 = Dataadc_3
Ldr_4 = Ldr_4
Locate 2 , 8
Lcd "" ; Fusing(ldr_4 , "#.#")
Lcd "     "
Locate 2 , 1
Lcd "LDR4 = "
If Ldr_4 > 100 Then
Eror4 = 1
Elseif Ldr_4 < 99 Then
Eror4 = 0
End If
End Sub
Sub Delay_1()
Incr Detik_1
If Detik_1 > 8 Then
Detik_1 = 0
Incr Detik
End If
End Sub
```

Data Sheet ATMega 8535

Block Diagram

Figure 2. Block Diagram



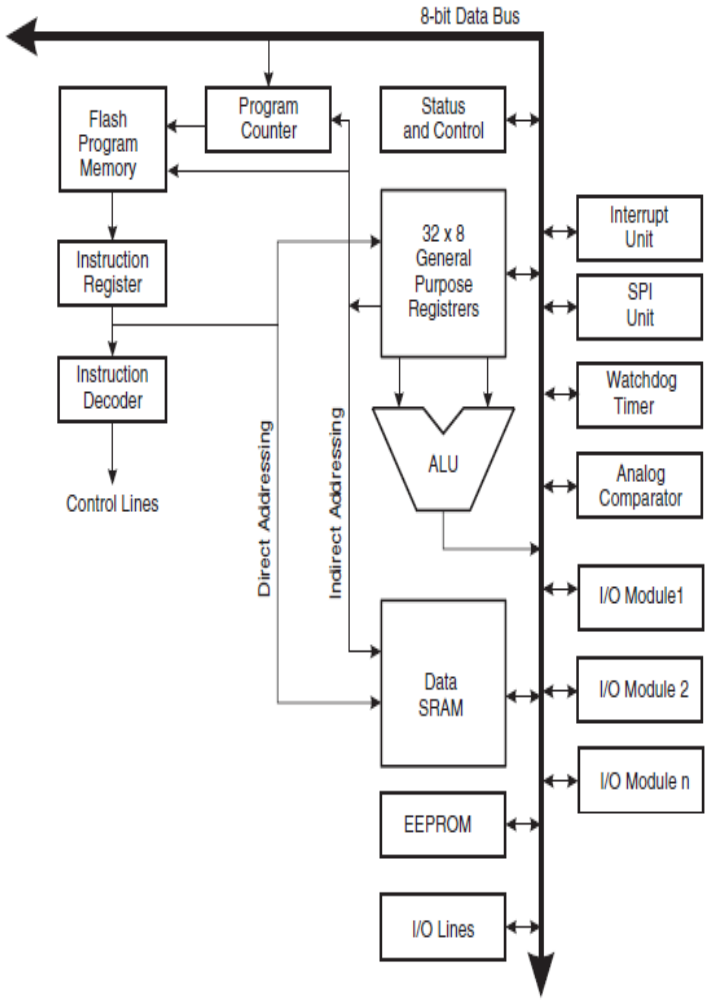
AVR CPU Core

Introduction

This section discusses the AVR core architecture in general. The main function of the CPU core is to ensure correct program execution. The CPU must therefore be able to access memories, perform calculations, control peripherals, and handle interrupts.

Architectural Overview

Figure 3. Block Diagram of the AVR MCU Architecture



In order to maximize performance and parallelism, the AVR uses a Harvard architecture – with separate memories and buses for program and data. Instructions in the program memory are executed with a single level pipelining. While one instruction is being executed, the next instruction is pre-fetched from the program memory. This concept enables instructions to be executed in every clock cycle. The program memory is In-System Re-Programmable Flash memory.

The fast-access Register File contains 32 x 8-bit general purpose working registers with a single clock cycle access time. This allows single-cycle Arithmetic Logic Unit (ALU) operation. In a typical ALU operation, two operands are output from the Register File, the operation is executed, and the result is stored back in the Register File – in one clock cycle.

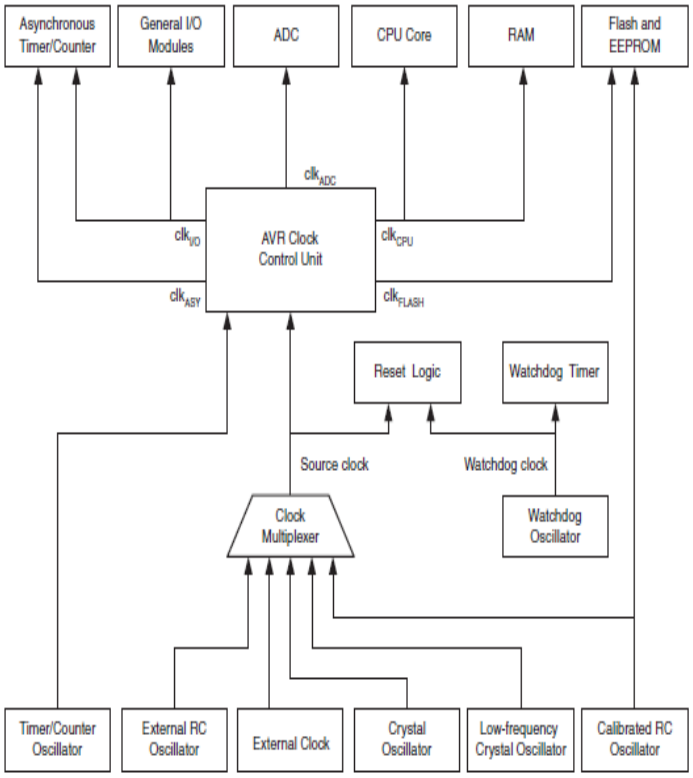
Universitas Medan Area

System Clock and Clock Options

Clock Systems and their Distribution

Figure 11 presents the principal clock systems in the AVR and their distribution. All of the clocks need not be active at a given time. In order to reduce power consumption, the clocks to modules not being used can be halted by using different sleep modes, as described in "Power Management and Sleep Modes" on page 32. The clock systems are detailed below.

Figure 11. Clock Distribution



CPU Clock – clk_{CPU}

The CPU clock is routed to parts of the system concerned with operation of the AVR core. Examples of such modules are the General Purpose Register File, the Status Register and the data memory holding the Stack Pointer. Halting the CPU clock inhibits the core from performing general operations and calculations.

I/O Clock – $clk_{I/O}$

The I/O clock is used by the majority of the I/O modules, like Timer/Counters, SPI, and USART. The I/O clock is also used by the External Interrupt module, but note that some external interrupts are detected by asynchronous logic, allowing such interrupts to be detected even if the I/O clock is halted. Also note that address recognition in the TWI module is carried out asynchronously when $clk_{I/O}$ is halted, enabling TWI address reception in all sleep modes.

Flash Clock – clk_{FLASH}

The Flash clock controls operation of the Flash interface. The Flash clock is usually active simultaneously with the CPU clock.

Figure 15. Reset Logic

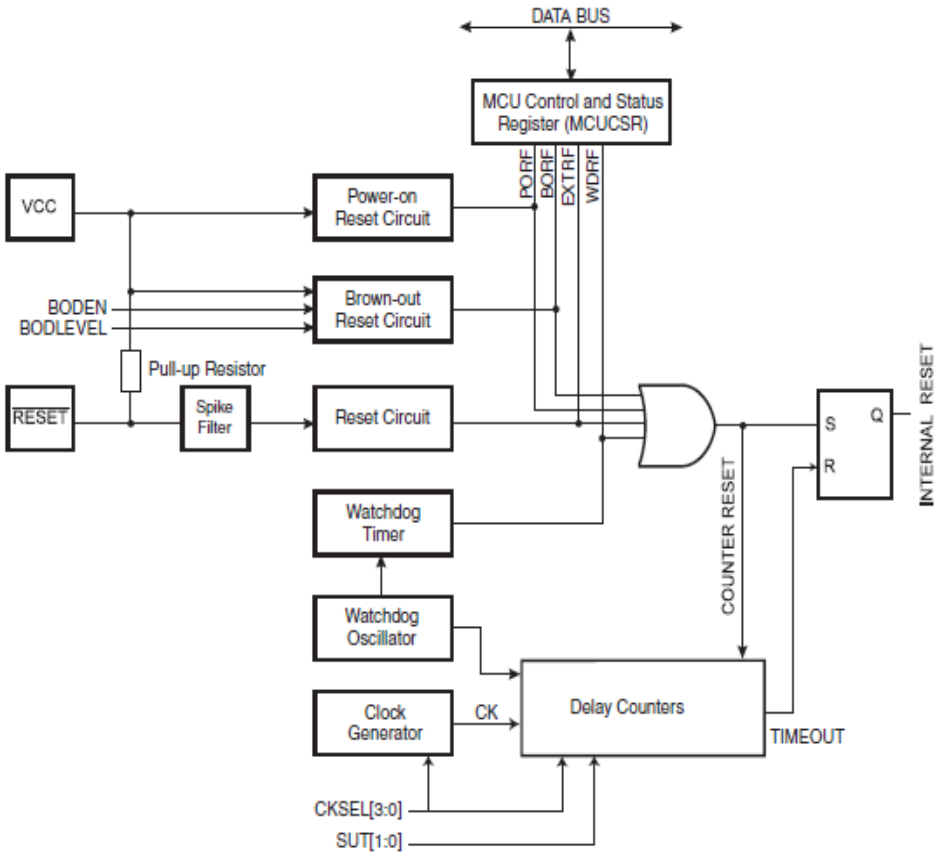


Table 15. Reset Characteristics

Symbol	Parameter	Condition	Min(1)	Typ(1)	Max(1)	Units
V_{POT}	Power-on Reset Threshold Voltage (rising)			1.4	2.3	V
	Power-on Reset Threshold Voltage (falling)(2)			1.3	2.3	V
V_{RST}	$\overline{\text{RESET}}$ Pin Threshold Voltage		0.2		0.9	V
t_{RST}	Minimum pulse width on $\overline{\text{RESET}}$ Pin				1.5	µs
V_{BOT}	Brown-out Reset Threshold Voltage(3)	BODLEVEL = 1	2.5	2.7	2.9	V
		BODLEVEL = 0	3.6	4.0	4.2	
t_{BOD}	Minimum low voltage period for Brown-out Detection	BODLEVEL = 1		2		µs
		BODLEVEL = 0		2		µs
V_{HYST}	Brown-out Detector hysteresis			130		mV

Notes: 1. Values are guidelines only.
2. The Power-on Reset will not work unless the supply voltage has been below V_{POT} (falling).

Interrupts

This section describes the specifics of the interrupt handling as performed in ATmega8535. For a general explanation of the AVR interrupt handling, refer to "Reset and Interrupt Handling" on page 13.

Interrupt Vectors in ATmega8535

Table 19. Reset and Interrupt Vectors

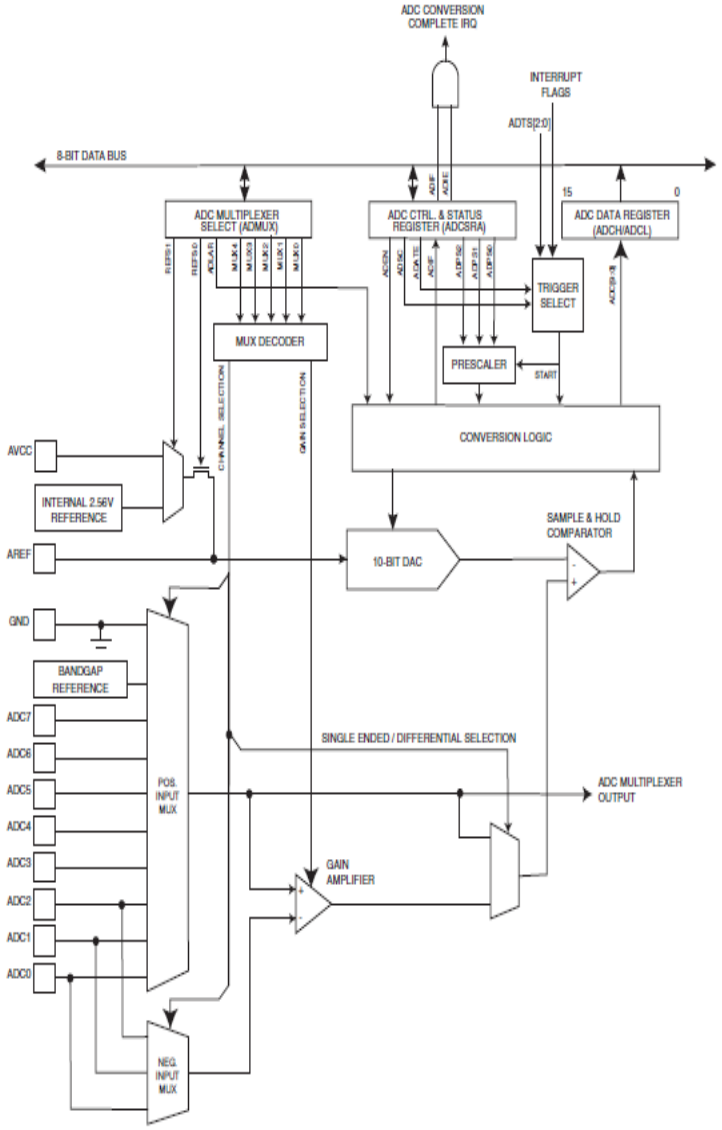
Vector No.	Program Address(2)	Source	Interrupt Definition
1	0x000(1)	RESET	External Pin, Power-on Reset, Brown-out Reset and Watchdog Reset
2	0x001	INT0	External Interrupt Request 0
3	0x002	INT1	External Interrupt Request 1
4	0x003	TIMER2 COMP	Timer/Counter2 Compare Match
5	0x004	TIMER2 OVF	Timer/Counter2 Overflow
6	0x005	TIMER1 CAPT	Timer/Counter1 Capture Event
7	0x006	TIMER1 COMPA	Timer/Counter1 Compare Match A
8	0x007	TIMER1 COMPB	Timer/Counter1 Compare Match B
9	0x008	TIMER1 OVF	Timer/Counter1 Overflow
10	0x009	TIMER0 OVF	Timer/Counter0 Overflow
11	0x00A	SPI, STC	Serial Transfer Complete
12	0x00B	USART, RXC	USART, Rx Complete
13	0x00C	USART, UDRE	USART Data Register Empty
14	0x00D	USART, TXC	USART, Tx Complete
15	0x00E	ADC	ADC Conversion Complete
16	0x00F	EE_RDY	EEPROM Ready
17	0x010	ANA_COMP	Analog Comparator
18	0x011	TWI	Two-wire Serial Interface
19	0x012	INT2	External Interrupt Request 2
20	0x013	TIMER0 COMP	Timer/Counter0 Compare Match
21	0x014	SPM_RDY	Store Program Memory Ready

Notes:
1. When the BOOTRST Fuse is programmed, the device will jump to the Boot Loader address at reset, see "Boot Loader Support – Read-While-Write Self-Programming" on page 224.
2. When the IVSEL bit in GICR is set, Interrupt Vectors will be moved to the start of the Boot Flash section. The address of each Interrupt Vector will then be the address in this table added to the start address of the Boot Flash section.

Table 20 shows reset and Interrupt Vectors placement for the various combinations of BOOTRST and IVSEL settings. If the program never enables an interrupt source, the Interrupt Vectors are not used, and regular program code can be placed at these locations. This is also the case if the Reset Vector is in the Application section while the Interrupt Vectors are in the Boot section or vice versa.

Universitas Medan Area

Figure 98. Analog-to-Digital Converter Block Schematic



Operation

The ADC converts an analog input voltage to a 10-bit digital value through successive approximation. The minimum value represents GND and the maximum value represents the voltage on the AREF pin minus 1 LSB. Optionally, AVCC or an internal 2.56V reference voltage may be connected to the AREF pin by writing to the REFSn bits in the ADMUX Register. The internal voltage reference may thus be decoupled by an external capacitor at the AREF pin to improve noise immunity.

The analog input channel and differential gain are selected by writing to the MUX bits in ADMUX. Any of the ADC input pins, as well as GND and a fixed bandgap voltage reference, can be selected as single ended inputs to the ADC. A selection of ADC input pins can be selected as positive and negative inputs to the differential gain amplifier.

If differential channels are selected, the differential gain stage amplifies the voltage difference between the selected input channel pair by the selected gain factor. This amplified value then becomes the analog input to the ADC. If single ended channels are used, the gain amplifier is bypassed altogether.

Memory Programming

Program And Data Memory Lock Bits

The ATmega8535 provides six Lock bits which can be left unprogrammed ("1") or can be programmed ("0") to obtain the additional features listed in Table 97. The Lock bits can only be erased to "1" with the Chip Erase command.

Table 96. Lock Bit Byte[1]

Lock Bit Byte	Bit No	Description	Default Value
	7	–	1 (unprogrammed)
	6	–	1 (unprogrammed)
BLB12	5	Boot Lock bit	1 (unprogrammed)
BLB11	4	Boot Lock bit	1 (unprogrammed)
BLB02	3	Boot Lock bit	1 (unprogrammed)
BLB01	2	Boot Lock bit	1 (unprogrammed)
LB2	1	Lock bit	1 (unprogrammed)
LB1	0	Lock bit	1 (unprogrammed)

Note: 1. "1" means unprogrammed, "0" means programmed

Table 97. Lock Bit Protection Modes[2]

Memory Lock Bits			Protection Type
LB Mode	**LB2**	**LB1**	
1	1	1	No memory lock features enabled.
2	1	0	Further programming of the Flash and EEPROM is disabled in Parallel and Serial Programming mode. The Fuse bits are locked in both Serial and Parallel Programming mode.[1]
3	0	0	Further programming and verification of the Flash and EEPROM is disabled in Parallel and Serial Programming mode. The Fuse bits are locked in both Serial and Parallel Programming mode.[1]
BLB0 Mode	**BLB02**	**BLB01**	
1	1	1	No restrictions for SPM or LPM accessing the Application section.
2	1	0	SPM is not allowed to write to the Application section.
3	0	0	SPM is not allowed to write to the Application section, and LPM executing from the Boot Loader section is not allowed to read from the Application section. If interrupt vectors are placed in the Boot Loader section, interrupts are disabled while executing from the Application section.
4	0	1	LPM executing from the Boot Loader section is not allowed to read from the Application section. If interrupt vectors are placed in the Boot Loader section, interrupts are disabled while executing from the Application section.
BLB1 Mode	**BLB12**	**BLB11**	

Universitas Medan Area

Table 97. Lock Bit Protection Modes[2] (Continued)

Memory Lock Bits			Protection Type
1	1	1	No restrictions for SPM or LPM accessing the Boot Loader section.
2	1	0	SPM is not allowed to write to the Boot Loader section.
3	0	0	SPM is not allowed to write to the Boot Loader section, and LPM executing from the Application section is not allowed to read from the Boot Loader section. If interrupt vectors are placed in the Application section, interrupts are disabled while executing from the Boot Loader section.
4	0	1	LPM executing from the Application section is not allowed to read from the Boot Loader section. If interrupt vectors are placed in the Application section, interrupts are disabled while executing from the Boot Loader section.

Notes: 1. Program the Fuse bits before programming the Lock bits.
2. "1" means unprogrammed, "0" means programmed.

Fuse Bits

The ATmega8535 has two Fuse bytes. Table 98 and Table 99 describe briefly the functionality of all the fuses and how they are mapped into the fuse bytes. Note that the fuses are read as logical zero, "0", if they are programmed.

Table 98. Fuse High Byte

Fuse High Byte	Bit No	Description	Default Value
S8535C	7	Select AT90S8535 compatibility mode	1 (unprogrammed)
WDTON	6	WDT always on	1 (unprogrammed, WDT enabled by WDTCR)
SPIEN[1]	5	Enable Serial Program and Data Downloading	0 (programmed, SPI prog. enabled)
CKOPT[2]	4	Oscillator options	1 (unprogrammed)
EESAVE	3	EEPROM memory is preserved through the Chip Erase	1 (unprogrammed, EEPROM not preserved)
BOOTSZ1	2	Select Boot Size (see Table 93 for details)	0 (programmed)[3]
BOOTSZ0	1	Select Boot Size (see Table 93 for details)	0 (programmed)[3]
BOOTRST	0	Select Reset Vector	1 (unprogrammed)

Notes: 1. The SPIEN Fuse is not accessible in Serial Programming mode.
2. The CKOPT Fuse functionality depends on the setting of the CKSEL bits. See "Clock Sources" on page 25. for details.
3. The default value of BOOTSZ1..0 results in maximum Boot Size. See Table 93 on page 235.

Universitas Medan Area

Table 99. Fuse Low Byte

Fuse Low Byte	Bit no	Description	Default Value
BODLEVEL	7	Brown out detector trigger level	1 (unprogrammed)
BODEN	6	Brown out detector enable	1 (unprogrammed, BOD disabled)
SUT1	5	Select start-up time	1 (unprogrammed)(1)
SUT0	4	Select start-up time	0 (programmed)(1)
CKSEL3	3	Select Clock source	0 (programmed)(2)
CKSEL2	2	Select Clock source	0 (programmed)(2)
CKSEL1	1	Select Clock source	0 (programmed)(2)
CKSEL0	0	Select Clock source	1 (unprogrammed)(2)

Notes: 1. The default value of SUT1..0 results in maximum start-up time. See Table 10 on page 30 for details.
2. The default setting of CKSEL3..0 results in internal RC Oscillator @ 1 MHz. See Table 2 on page 25 for details.

The status of the Fuse bits is not affected by Chip Erase. Note that the Fuse bits are locked if Lock bit 1 (LB1) is programmed. Program the Fuse bits before programming the Lock bits.

Latching of Fuses

The Fuse values are latched when the device enters Programming mode and changes of the Fuse values will have no effect until the part leaves Programming mode. This does not apply to the EESAVE fuse which will take effect once it is programmed. The fuses are also latched on Power-up in Normal mode.

Signature Bytes

All Atmel microcontrollers have a three-byte signature code which identifies the device. This code can be read in both Serial and Parallel mode, also when the device is locked. The three bytes reside in a separate address space.

For the ATmega8535 the signature bytes are:

1. 0x000: 0x1E (indicates manufactured by Atmel)
2. 0x001: 0x93 (indicates 8 KB Flash memory)
3. 0x002: 0x08 (indicates ATmega8535 device when 0x001 is 0x93)

Calibration Byte

The ATmega8535 stores four different calibration values for the internal RC Oscillator. These bytes resides in the signature row high byte of the addresses 0x000, 0x0001, 0x0002, and 0x0003 for 1, 2, 4, and 8 MHz respectively. During Reset, the 1 MHz value is automatically loaded into the OSCCAL Register. If other frequencies are used, the calibration value has to be loaded manually, see "Oscillator Calibration Register – OSCCAL" on page 30 for details.

Universitas Medan Area

Electrical Characteristics

Absolute Maximum Ratings*

Operating Temperature	-55°C to +125°C
Storage Temperature	-65°C to +150°C
Voltage on any Pin except $\overline{\text{RESET}}$ with respect to Ground	-0.5V to V_{CC}+0.5V
Voltage on $\overline{\text{RESET}}$ with respect to Ground	-0.5V to +13.0V
Maximum Operating Voltage	6.0V
DC Current per I/O Pin	40.0 mA
DC Current V_{CC} and GND Pins	200.0 PDIP og
400 mA TQFP/MLF/PLCCmA	

*NOTICE: Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC Characteristics

T_A = -40°C to 85°C, V_{CC} = 2.7V to 5.5V (unless otherwise noted)

Symbol	Parameter	Condition	Min	Typ	Max	Units
V_{IL}	Input Low Voltage except XTAL1 and $\overline{\text{RESET}}$ pins	V_{CC}=2.7V - 5.5V	-0.5		0.2 V_{CC}[1]	V
V_{IH}	Input High Voltage except XTAL1 and $\overline{\text{RESET}}$ pins	V_{CC}=2.7V - 5.5V	0.6 V_{CC}[2]		V_{CC} + 0.5	V
V_{IL1}	Input Low Voltage XTAL1 pin	V_{CC}=2.7V - 5.5V	-0.5		0.1 V_{CC}[1]	V
V_{IH1}	Input High Voltage XTAL1 pin	V_{CC}=2.7V - 5.5V	0.8 V_{CC}[2]		V_{CC} + 0.5	V
V_{IL2}	Input Low Voltage $\overline{\text{RESET}}$ pin	V_{CC}=2.7V - 5.5V	-0.5		0.2 V_{CC}	V
V_{IH2}	Input High Voltage $\overline{\text{RESET}}$ pin	V_{CC}=2.7V - 5.5V	0.9 V_{CC}[2]		V_{CC} + 0.5	V
V_{OL}	Output Low Voltage[3] (Ports A,B,C,D)	I_{OL} = 20 mA, V_{CC} = 5V I_{OL} = 10 mA, V_{CC} = 3V			0.7 0.5	V V
V_{OH}	Output High Voltage[4] (Ports A,B,C,D)	I_{OH} = -20 mA, V_{CC} = 5V I_{OH} = -10 mA, V_{CC} = 3V	4.2 2.2			V V
I_{IL}	Input Leakage Current I/O Pin	V_{CC} = 5.5V, pin low (absolute value)			1	µA
I_{IH}	Input Leakage Current I/O Pin	V_{CC} = 5.5V, pin high (absolute value)			1	µA
R_{RST}	Reset Pull-up Resistor		30		60	kΩ
R_{pu}	I/O Pin Pull-up Resistor		20		50	kΩ

Universitas Medan Area

ATmega8535 Typical Characteristics

The following charts show typical behavior. These figures are not tested during manufacturing. All current consumption measurements are performed with all I/O pins configured as inputs and with internal pull-ups enabled. A sine wave generator with rail-to-rail output is used as clock source.

The power consumption in Power-down mode is independent of clock selection.

The current consumption is a function of several factors such as: Operating voltage, operating frequency, loading of I/O pins, switching rate of I/O pins, code executed and ambient temperature. The dominating factors are operating voltage and frequency.

The current drawn from capacitive loaded pins may be estimated (for one pin) as $C_L * V_{CC} * f$ where C_L = load capacitance, V_{CC} = operating voltage and f = average switching frequency of I/O pin.

The parts are characterized at frequencies higher than test limits. Parts are not guaranteed to function properly at frequencies higher than the ordering code indicates.

The difference between current consumption in Power-down mode with Watchdog Timer enabled and Power-down mode with Watchdog Timer disabled represents the differential current drawn by the Watchdog Timer.

Active Supply Current

Figure 130. Active Supply Current vs. Frequency (0.1 - 1.0 MHz)

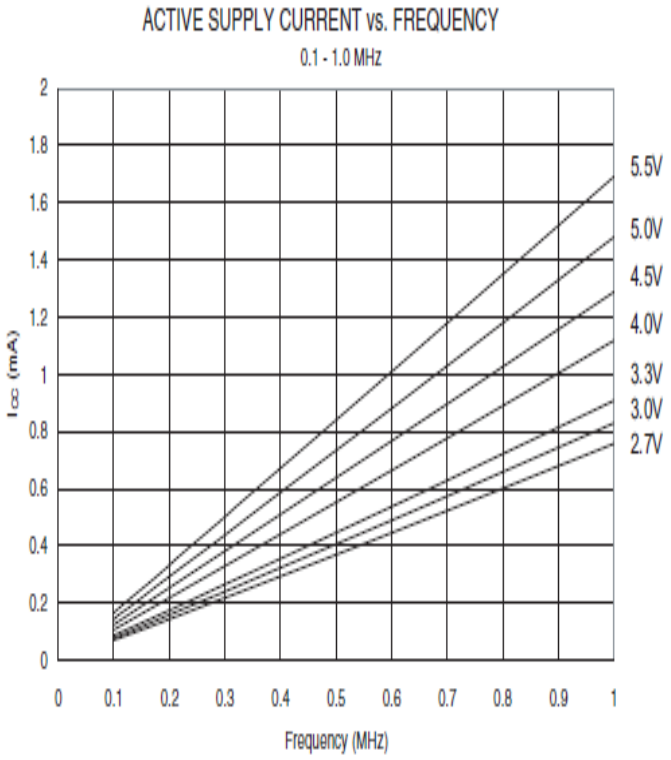


Figure 131. Active Supply Current vs. Frequency (1 - 16 MHz)

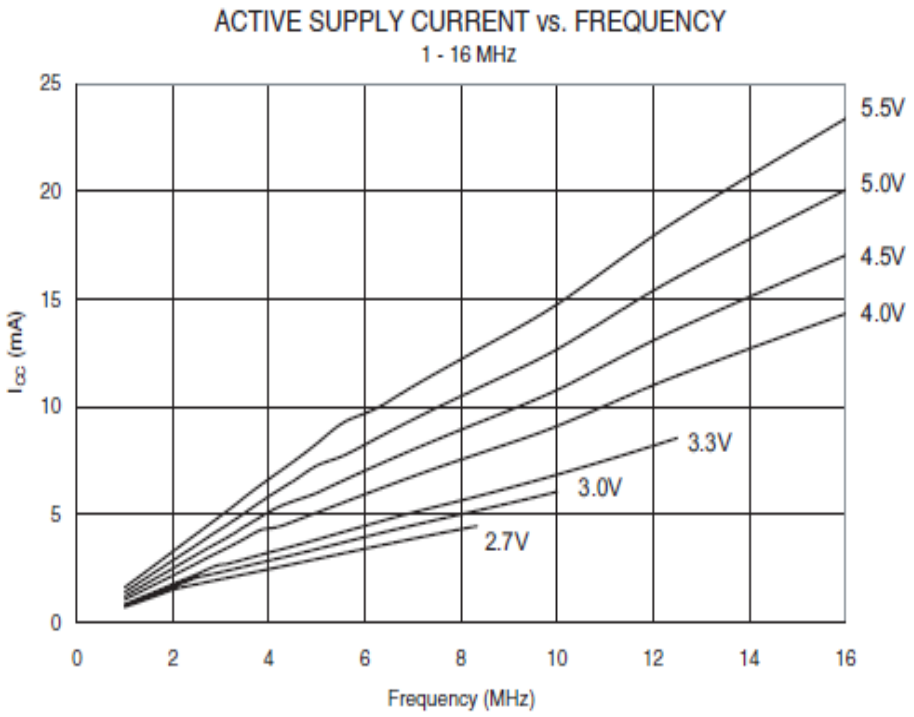


Figure 132. Active Supply Current vs. V_{CC} (Internal RC Oscillator, 8 MHz)