

Vážení zákazníci,

dovolujeme si Vás upozornit, že na tuto ukázku knihy se vztahuje autorská práva, tzv. copyright.

To znamená, že ukázka má sloužit výhradně pro osobní potřebu potenciálního kupujícího (aby čtenář viděl, jakým způsobem je titul zpracován a mohl se také podle tohoto, jako jednoho z parametrů, rozhodnout, zda titul koupí či ne).

Z toho vyplývá, že není dovoleno tuto ukázku jakýmkoliv způsobem dále šířit, veřejně či neveřejně např. umisťováním na datová média, na jiné internetové stránky (ani prostřednictvím odkazů) apod.

AVR – ŘADY MIKROKONTROLÉRŮ

Atmel vyrábí mikrokontroléry AVR ve třech řadách:

Základní

Obsahuje typy AT90S1200, AT90S2313, AT90S2323, AT90S2343, AT90S4433, AT90S4434, AT90S8515, AT90S8534 a AT90S8535. Tyto typy (s výjimkou AT90S1200) mají 118 instrukcí a jsou u nás dostupné v maloobchodní síti. Typ AT90S1200 má 89 instrukcí a oproti dalším typům nemá paměť SRAM. Proto se k napsání sw pro tento typ nedají použít některé překladače vyšších jazyků. Na druhé straně výhodou tohoto typu je jeho nízká cena.

ATtiny

Obsahuje typy ATtiny11, ATtiny12, ATtiny15, ATtiny28 a ATtiny26. S výjimkou posledního typu ATtiny26, který má 118 instrukcí, mají ostatní 90 instrukcí. Rovněž typy této rodiny jsou u nás v maloobchodě dostupné. Jsou to nejlacinější typy mikrokontrolérů AVR.

ATmega

Obsahuje jednak starší řadu ATmega103, ATmega161, ATmega163 a ATmega323 a novější řadu ATmega8, ATmega16, ATmega64 a ATmega128. Tyto mikrokontroléry mají (s výjimkou ATmega103) 130 instrukcí. Oproti základní řadě obsahuje navíc instrukce pro násobení. Novější řada je navíc vybavena rozhraním JTAG pro ladění sw přímo v aplikaci. Většina novinek v oblasti AVR se týká právě řady ATmega a to jak nové typy mikrokontrolérů, tak novinky v oblasti vývojových prostředků.

Při získávání prvních praktických zkušeností s ATMELOM AVR MCU pravděpodobně použijeme to, co je v současné době k dispozici v maloobchodních prodejnách, tedy MCU základní řady. Jejich stručný popis je obsahem *tab. 2.1*.

Tab. 2.1 Vlastnosti MCU základní řady

	1200	2313	2343	4433	8515	8535
Max. kmitočet CPU v [MHz]	4 popř. 12	4 popř. 10	1; 4; 10	4 popř. 8	4 popř. 8	4 popř. 8
Počet instrukcí	89	118	118	118	118	118
Počet obecných registrů	32	32	32	32	32	32
FLASH	1kB	2kB	2kB	4kB	8kB	8kB
SRAM	-	128B	128B	128B	512B	512B
EEPROM	64B	128B	128B	256B	512B	512B
Čítač/časovač0	8 bit					
Čítač/čacovač1	-	16 bit	-	16 bit	16 bit	16 bit
Čítač/časovač2	-	-	-	-	-	8 bit
Watchdog	ano	ano	ano	ano	ano	ano
A/D převodník	-	-	-	10 bit	-	10 bit
Analog. komp.	ano	ano	ano	ano	ano	ano
UART	-	ano	ano	ano	ano	ano
SPI sér. interface	ano	ano	ano	ano	ano	ano
PORT A	-	-	-	-	ano	ano
PORT B	ano	ano	ano	ano	ano	ano
PORT C	-	-	-	ano	ano	ano
PORT D	ano	ano	-	ano	ano	ano

6

I/O PORTY

MCU řady AT90 v základní řadě mohou být vybaveny až čtyřmi 8bitovými obousměrnými branami PORT A, PORT B, PORT C a PORT D. U řady ATmega mohou být ještě PORT E a PORT F. Brány jsou mapovány každá na tři adresy – vstupní vývody, výstupní registr a směrový registr. Jelikož základní funkce těchto tří registrů je stejná u všech portů, bran, popíšeme si jen registry pro PORT A.

Port A Inputs Pins adresa (PINa) není registr, tato adresa umožňuje přístup

Port A Data Register - PORTA

datový registr PORTu A

bit	7	6	5	4	3	2	1	0	
\$1B (\$3B)	PORTA7	PORTA6	PORTA5	PORTA4	PORTA3	PORTA2	PORTA1	PORTA0	PORTA
čtení/zápis	R/W								
počáteční hodnota	0	0	0	0	0	0	0	0	

Port A Data Direction Register - DDRA

registrování směru přenosu dat PORTu A

bit	7	6	5	4	3	2	1	0	
\$1A (\$3A)	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	DDRA
čtení/zápis	R/W								
počáteční hodnota	0	0	0	0	0	0	0	0	

Port A Input Pins Address - PINA

adresa vstupních pinů PORTu A

bit	7	6	5	4	3	2	1	0	
\$19 (\$39)	PINA7	PINA6	PINA5	PINA4	PINA3	PINA2	PINA1	PINA0	PINA
čtení/zápis	R	R	R	R	R	R	R	R	
počáteční hodnota	N/A								

Obr. 6.1 Registry PORTu A

k fyzické hodnotě vlastního portu, na adrese vstupních/výstupních vývodů jsou příslušné skutečné úrovně na vývodech integrovaného obvodu, na adresu výstupního registru lze zapisovat výstupní data. Data zapsaná do směrového registru definují funkci příslušného vývodu. Vliv dat zapsaných do registru dat a směrového registru na chování vývodu shrnuje tab. 6.1.

Všechny bity v režimu výstupu mohou být v log. 0 zatíženy proudem až 20 mA.

Tab. 6.1 Stav vývodu

DDAn	PORTAn	I/O	Stav vývodu
0	0	Vstup	Velká impedance (otevřený kolektor v log.1)
0	1	Vstup	Připojen zatěžovací odpor, vstup může být zdrojem proudu
1	0	Výstup	Výstup v log.0, otevřený kolektor
1	1	výstup	Výstup v log.1, otevřený kolektor

Přístup na I/O porty je základní dovednost nezbytně nutná pro jakoukoliv činnost MCU. AVR umožňují přístup bitový i bytový. Pro bitový přístup k portu se využívají dva registry. Pro čtení jednotlivých bitů se používají např. instrukce SBIC a SBIS (např. SBIS PINA, 4), pro zápis např. instrukce SBI a CBI (např. SBI PORTA, 0). Pro bytový přístup používáme instrukce IN a OUT (např. IN DATA, PINA).

Jednotlivé byty bran mohou mít přiřazeny alternativní funkce nastavením konfiguračních bitů v registrech příslušných periferií, popř. v kombinaci s nastavením bitů ve směrovém registru brány. Alternativní funkce jsou přiřazeny podle typu pouzdra a MCU, nastavení nutné pro výkon těchto funkcí je však u všech typů shodné.

Kromě vstupní/výstupních bran, portů, jsou součásti I/O prostoru i stavový registr, či registry související s prací vestavěných periferií. Proto si teď stručně popíšeme funkci zbývajících registrů I/O prostoru:

Stavový registr – SREG

AVR stavový registr zaujímá v I/O prostoru pozici \$3F (\$5F)

Status Register - SREG								
bit	7	6	5	4	3	2	1	0
\$3F (\$5F)	I	T	H	S	V	N	Z	C
čtení/zápis	R/W							
počáteční hodnota	0	0	0	0	0	0	0	0

SREG

Obr. 6.2 Stavový registr

Bit 7 – Global Interrupt Enable – povolení všech přerušení

Tento bit musí být nastaven na jedničku, aby povoloval všechna přerušení. Povolení jednotlivých přerušení se provádí v příslušných oddělených řídicích registrech. Je-li bit I vynulován, jsou zakázána všechna přerušení, nezáleží přitom na případných povolení přerušení v individuálních řídicích registrech. Bit I je nulován hardwarem poté, co dojde k přerušení. Nastavován je instrukcí RETI.

Bit 6 – T: Bit Copy Storage, příznak Transfer Bit

Instrukce provádějící kopírování bitu BLD (BitLoaD) a BST (Bit Store) používají bit T jako zdrojový či cílový bit.

Bit 5 – H: Half-carry Flag

Přenos mezi třetím a čtvrtým bitem (využíván BCD aritmetikou).

Bit 4/S: Sign Bit, S = N \oplus V

Tento bit je výlučným součtem (exclusive or) mezi příznakem záporného výsledku N a příznakem přetečení V, to znamená, že určuje znaménko výsledku.

Bit 3 – V: Two's Complement Overflow Flag

Příznak přetečení dvojkového doplňku.

Bit 2 – N: Negative Flag

Příznak záporného výsledku aritmetických a logických operací.

Bit 1 – Z: Zero Flag

Příznak Z indikuje nulový výsledek aritmetických nebo logických operací.

Bit 0 – C: Carry Flag

Příznak C indikuje přenos při aritmetických a logických operacích

Ukazatel na zásobník SP (Stack Pointer)

Ukazatel zásobníku je implementován jako dva 8bitové registry v I/O prostoru. Počet použitých bitů závisí na typu MCU. Uvedeme si SP pro AT90S8535

Stack Pointer SP
ukazatel na zásobník

bit	15	14	13	12	11	10	9	8	
\$3E (\$5E)	—	—	—	—	—	—	SP9	SP8	SPH
\$3D (\$5D)	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	SPL
	7	6	5	4	3	2	1	0	
	R	R	R	R	R	R	R	R	
čtení/zápis	R/W								
počáteční hodnota	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

Obr. 6.3 Ukazatel na zásobník

Ukazatel SP ukazuje na zásobník umístěný v SRAM. Prostor zásobníku v SRAM musí být definován programem před jeho používáním voláním podprogramů či obsluhy přerušení. Musí být nastaven nad \$60 (důvod je zřejmý z obr. 1.2 – nižší adresy nepatří SRAM).

PUSH

PUSH REGISTER ON STACK

Uložení do zásobníku

Popis:

Tato instrukce ukládá obsah registru Rd do zásobníku. Ukazatel na zásobník je post dekrementován o jedničkou po provedení PUSH.

Syntaxe instrukce:

PUSH Rd

Operandy:

$0 \leq d \leq 31$

Operace:

STACK \leftarrow Rd

Programový čítač:

PC \leftarrow PC + 1

Ukazatel zásobníku:

SP \leftarrow SP - 1

Šestnáctibitový kód instrukce:

1001	001d	dddd	1111
------	------	------	------

Stavový registr (SREG):

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	-	-

RCALL

RELATIVE CALL TO SUBROUTINE

Relativní volání podprogramu

Popis:

Relativní volání na adresu v rozsahu PC –2K + 1 a PC + 2K (slov). Návratová adresa (instrukce po RCALL) je zapamatována v zásobníku (viz také CALL). V assembleru, jsou labely (návěští) užity místo relativních operandů. Pro AVR mikrokontroléry s programovou pamětí nepřekračující 4K slov (8K bytů) může tato instrukce adresovat celou paměť z libovolné adresové pozice. Ukazatel zásobníku užívá post dekrementační postup během RCALL.

Syntaxe instrukce:

RCALL k

Operandy:

$-2K \leq k \leq 2K$

Operace:

- (i) $PC \leftarrow PC + k + 1$ MCU s 16bitovým PC, max. 128K bytů paměti.
- (ii) $PC \leftarrow PC + k + 1$ MCU s 22bitovým PC, max. 8M bytů paměti.

Zásobník:

- (i) $STACK \leftarrow PC + 1$ $SP \leftarrow SP - 2$ (2 byty, 16 bitů)
- (ii) $STACK \leftarrow PC + 1$ $SP \leftarrow SP - 3$ (3 byty, 22 bitů)

Programový čítač:

$PC \leftarrow PC + k + 1$

Šestnáctibitový kód instrukce:

1101	kkkk	kkkk	kkkk
------	------	------	------

RET

RETURN FROM SUBROUTINE

Návrat z podprogramu

Popis:

Návrat z podprogramu. Návratová adresa je přesunuta ze zásobníku. Ukazatel na zásobník používá pre inkrementační postup během RET.

Syntaxe instrukce:

RET

Operandy:

Žádné

Operace:

- (i) PC (15:0) ← STACK MCU s 16bitovým PC, max. 128K programové paměti
- (ii) PC (21:0) ← STACK MCU s 22bitovým PC, max. 8M programové paměti

Programový čítač:

Viz operace.

Zásobník:

- (i) SP ← SP + 2 (2 byty, 16 bitů)
- (ii) SP ← SP + 3 (3 byty, 22 bitů)

Šestnáctibitový kód instrukce:

1001	0101	0000	1000
------	------	------	------

SBR

SET BITS IN REGISTER

Nastavuje bity v registru

Popis:

Nastavuje specifikované bity v registru Rd. Vytváří logický ORI mezi obsahem registru Rd a obsahem konstantní masky K a umisťuje výsledek do cílového registru Rd. SBR je jen jiné značení instrukce ORI – obě mají stejný kód instrukce.

Syntaxe instrukce:

SBR Rd, K

Operandy:

16 ≤ d ≤ 31, 0 ≤ K ≤ 255

Operace:

$Rd \leftarrow Rd \vee K$

Programový čítač:

$PC \leftarrow PC + 1$

Šestnáctibitový kód instrukce:

0110	KKKK	dddd	KKKK
------	------	------	------

tento kód je stejný jako kód instrukce ORI.

Stavový registr (SREG):

I	T	H	S	V	N	Z	C
-	-	-	↔	0	↔	↔	-

SER

SET ALL BITS IN REGISTER

Nastavuje všechny bity v registru

Popis:

Přesouvá \$FF přímo do registru Rd.

Syntaxe instrukce:

SER Rd

Operandy:

$16 \leq d \leq 31$

Operace:

$Rd \leftarrow \$FF$

Programový čítač:

$PC \leftarrow PC+1$

Šestnáctibitový kód instrukce:

1110	1111	dddd	1111
------	------	------	------

Stavový registr (SREG):

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	-	-

Počet slov instrukce:

1 (2 byte)

SES

SET **S**IGNED **F**LAG

Nastavuje znaménkový příznak

Popis:

Nastavuje znaménkový příznak (S) v SREG (stavový registr).

Syntaxe instrukce:

SES

Operandy:

Žádný

Operace:

$S \leftarrow 1$

Programový čítač:

$PC \leftarrow PC + 1$

Šestnáctibitový kód instrukce:

1001	0100	0100	1000
------	------	------	------

Stavový registr (SREG):

I	T	H	S	V	N	Z	C
-	-	-	1	-	-	-	-

S: 1

nastavení příznaku znaménka.