

**ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA ELEKTROTECHNICKÁ**

KATEDRA ELEKTROTECHNIKY A VÝKONOVÉ ELEKTRONIKY

DIPLOMOVÁ PRÁCE

Vývojový modul s mikroprocesorem ADuC8xx

vedoucí práce: Ing. Zuzana Petránková, Ph.D.
autor: Bc. Jaroslav Vávra

2014

ZÁPADOČESKÁ UNIVERZITA V PLZNI
Fakulta elektrotechnická
Akademický rok: 2013/2014

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: Bc. Jaroslav VÁVRA
Osobní číslo: E12N0032K
Studijní program: N2644 Aplikovaná elektrotechnika
Studijní obor: Aplikovaná elektrotechnika
Název tématu: Vývojový modul s mikroprocesorem ADuC8xx
Zadávající katedra: Katedra elektroenergetiky a ekologie

Rozsah grafických prací: podle doporučení vedoucího
Rozsah pracovní zprávy: 30 - 40 stran
Forma zpracování diplomové práce: tištěná/elektronická
Seznam odborné literatury:

Student si vhodnou literaturu vyhledá v dostupných pramenech podle doporučení vedoucího práce.

Zásady pro vypracování:

1. Prostudujte materiály o obvodech ADuC8xx.
2. Navrhněte univerzální vývojový modul s vybraným mikrokontrolérem ADuC8xx.
3. Navrhněte komunikační a zobrazovací rozhraní.
4. Realizujte funkční vzorek.
Vytvořte SW k ověření funkce Vámi navrženého HW.
5. Výsledky v práci podrobně popište.

Vedoucí diplomové práce: Ing. Zuzana Petránková, Ph.D.
Katedra aplikované elektroniky a telekomunikací

Datum zadání diplomové práce: 14. října 2013
Termín odevzdání diplomové práce: 12. května 2014




Doc. Ing. Jiří Hammerbauer, Ph.D.
děkan




Doc. Ing. Karel Noháč, Ph.D.
vedoucí katedry

V Plzni dne 14. října 2013



Anotace

Předkládaná diplomová práce se věnuje problematice mikrokontrolérů řady ADuC8xx od firmy . Tyto mikrokontroléry jsou kompatibilní s mikrokontroléry řady INTEL 8051 a s jejich nástupcem INTEL 8052. Úvodní část je zaměřena na seznámení s problematikou mikrokontroléru ADuC8xx. V rámci práce byla provedena podrobná rešerše této problematiky na základě katalogových listů mikrokontrolérů řady ADuC8xx od firmy . Ve vlastní části práce bylo pro vývojový modul, navržený v rámci méj bakalářské práce, vytvořeno několik ukázkových programů a aplikací.

Klíčová slova

INTEL 8051, INTEL 8052, Analog devices, mikrokontrolér, vývojový modul, ADuC812, ADuC814, ADuC816, ADuC824, ADuC831, ADuC832, ADuC834, ADuC836, ADuC841, ADuC842, ADuC843, ADuC845, ADuC847, ADuC848

Abstract

Presented thesis deals with the issue of microcontroller series ADuC8xx produced by company . These microcontrollers are compatible with the INTEL 8051 family of microcontrollers and their successor, the INTEL 8052. The introductory part focuses on an introduction of microcontroller ADuC8xx. In the framework of the work the detailed examination of this issue on the basis of the catalog sheets of microcontroller series ADuC8xx from the company  has been carried out. The second part of the thesis deals with the development of several typical program and applications designed for the developmental module developed in the framework of thesis.

Key words

INTEL 8051, INTEL 8052, Analog devices, microcontroller, development module, ADuC812, ADuC814, ADuC816, ADuC824, ADuC831, ADuC832, ADuC834, ADuC836, ADuC841, ADuC842, ADuC843, ADuC845, ADuC847, ADuC848

Prohlášení

Předkládám tímto k posouzení a obhajobě diplomovou práci, zpracovanou na závěr studia na Fakultě elektrotechnické Západočeské univerzity v Plzni.

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně, s použitím odborné literatury a pramenů uvedených v seznamu, který je součástí této diplomové práce.

Dále prohlašuji, že veškerý software, který byl použitý při řešení této diplomové práce, je legální.

V Plzni dne 8. května 2014

.....
Bc. Jaroslav Vávra

Poděkování

Chtěl bych tímto poděkovat všem, kteří mi byli přímo, ale i nepřímo nápomocni při přípravě a realizaci této diplomové práce. Vedoucímu diplomové práce Ing. Zuzaně Petránkové, Ph.D. děkuji za cenné profesionální rady, připomínky a metodické vedení práce. Doc. Dr. Ing. Vjačeslavu Georgievovi za cenné rady v oblasti režimů nízké spotřeby. Dále bych chtěl poděkovat rodině a hlavně manželce a dětem za podporu a pochopení v průběhu celého studia.

Obsah

SEZNAM SYMBOLŮ A ZKRATEK.....	9
ÚVOD.....	10
1. ARCHITEKTURA MIKROKONTROLÉRŮ ADUC8XX.....	11
1.1. ORGANIZACE PAMĚTI.....	11
1.1.1. Paměť programu.....	11
1.1.2. Paměť dat.....	12
1.2. INTEGROVANÉ UŽIVATELSKÉ FUNKČNÍ BLOKY.....	13
1.2.1. Paměť Flash/EE.....	13
1.2.2. A/D převodník.....	13
1.2.3. D/A převodník.....	13
1.2.4. Pulsně šířkový modulátor PWM.....	13
1.2.5. SPI komunikace.....	17
1.2.6. I2C kompatibilní rozhraní.....	18
1.2.7. Monitor napájení.....	18
1.2.8. Hlídač běhu (Watchdog timer).....	18
1.2.9. Časový čítač (TIC).....	19
2. VÝVOJOVÝ MODUL.....	20
2.1. NAHRÁVÁNÍ PROGRAMU DO MIKROKONTROLÉRU.....	22
2.2. EXTERNÍ PERIFERIE A UKÁZKOVÝ PROGRAM.....	23
2.2.1. Režim se sníženou spotřebou.....	24
2.2.2. Snímač teploty s analogovým výstupem (TMP36).....	28
2.2.3. Interní časovač jako synchronizace běhu programu.....	31
2.2.4. Displej 4x20 znaků (FDCC2004C – FLYYBW – 51LK).....	33
2.2.5. Pulsně šířková modulace (PWM).....	39
ZÁVĚR.....	40
POUŽITÁ LITERATURA.....	41
SEZNAM OBRÁZKŮ.....	42
SEZNAM TABULEK.....	42
SEZNAM PŘÍLOH.....	43
PŘÍLOHA A – BLOKOVÉ SCHÉMA ADUC8XX.....	45
PŘÍLOHA B – SPECIAL FUNCTION REGISTERS (SFR) OBLAST SPECIÁLNÍCH REGISTRŮ....	48
PŘÍLOHA C – PARAMETRY SFR ADUC8XX.....	54
C.1. ADC0CON (PRIMARY ADC CONTROL REG. – ŘÍDÍCÍ SLOVO PRIMÁRNÍHO A/D).....	54
C.2. ADC0H/M/L (PRIMARY ADC CONVERSION RESULT REG. – DATA Z PRIM. A/D).....	54
C.3. ADC1CON (AUXILIARY ADC CONTROL REG. – ŘÍDÍCÍ SLOVO POMOCNÉHO A/D).....	54
C.4. ADC1H/L (AUXILIARY ADC CONVERSION RESULT REG. – DATA Z POMOCN. A/D).....	55
C.5. ADCCON1 (ADC CONTROL – ŘÍDÍCÍ SLOVO 1 A/D PŘEVODNÍKU).....	55
C.6. ADCCON2 (ADC CONTROL – ŘÍDÍCÍ SLOVO 2 A/D PŘEVODNÍKU).....	56
C.7. ADCCON3 (ADC CONTROL – ŘÍDÍCÍ SLOVO 3 A/D PŘEVODNÍKU).....	58
C.8. ADCDATAH/L (ADC DATA REGISTERS – DATA Z A/D PŘEVODNÍKU).....	58
C.9. ADCGAINH/L (ADC GAIN CALIBRATIN COEFFICIENT – ZISK A/D PŘEVODNÍKU).....	58
C.10. ADCMODE (ADC MODE REGISTER – REŽIM A/D PŘEVODNÍKU).....	59
C.11. ADCOFSH/L (ADC OFFSET CALIBRATIN COEFFICIENT – POSUN A/D PŘEVODNÍKU).....	59
C.12. ADCSTAT (ADC STATUS REGISTER – STAVOVÝ REGISTR).....	59
C.13. CFG814 (ADUC814 CONFIGURATION REGISTER – KONFIGURACE).....	60


C.14. CFG831 (ADuC831 CONFIGURATION REGISTER – KONFIGURACE).....	60
C.15. CFG832 (ADuC832 CONFIGURATION REGISTER – KONFIGURACE).....	61
C.16. CFG834 (ADuC834 CONFIGURATION REGISTER – KONFIGURACE).....	61
C.17. CFG836 (ADuC836 CONFIGURATION REGISTER – KONFIGURACE).....	61
C.18. CFG841 (ADuC841 CONFIGURATION REGISTER – KONFIGURACE).....	62
C.19. CFG842 (ADuC842 CONFIGURATION REGISTER – KONFIGURACE).....	62
C.20. DACCON (DAC CONTROL – ŘÍDÍCÍ SLOVO D/A PŘEVODNÍKU).....	63
C.21. DACH/L (DAC DATA REG. – DATA DO D/A PŘEVODNÍKU).....	63
C.22. DACxH/L (DAC DATA REG. – DATA DO D/A PŘEVODNÍKU).....	64
C.23. DMAL/H/P (DMA ADDRESS POINTER – UKAZATEL DMA TABULKY).....	64
C.24. ECON (FLASH/EE DATA SPACE CONTROLS – ŘÍDÍCÍ SLOVO).....	64
C.25. EDARL/(H) (FLASH/EE DATA PAGE ADDRESS – STRÁNKA FLASH/EE).....	65
C.26. EDATA1 – 4 (FLASH/EE DATA).....	65
C.27. ETIM1 – 3 (FLASH/EE MEMORY TIMIG).....	65
C.28. GN0H/M/(L) (PRIMARY ADC GAIN CALIBRATION – ZISK PRIMÁRNÍHO A/D).....	65
C.29. GN1H/L (AUXILIARY ADC GAIN CALIBRATION – ZISK POMOCNÉHO A/D).....	65
C.30. HOUR (HOURS TIME REGISTER – REGISTR HODIN).....	66
C.31. HTHSEC (HUNDREDTHS SECONDS TIME REGISTER – REGISTR SETIN VTEŘINY).....	66
C.32. I2CADD (I2C ADDRESS REGISTER – I2C ADRESA).....	66
C.33. I2CADD1 (I2C ADDRESS REGISTER – I2C ADRESA).....	66
C.34. I2CADD2 (I2C ADDRESS REGISTER – I2C ADRESA).....	66
C.35. I2CADD3 (I2C ADDRESS REGISTER – I2C ADRESA).....	66
C.36. I2CCON (I2C CONTROL REGISTER – ŘÍDÍCÍ SLOVO I2C).....	67
C.37. I2CDAT (I2C DATA REGISTER – DATA PRO I2C PŘENOS).....	68
C.38. IE (INTERRUPT ENABLE – POVOLENÍ PŘERUŠENÍ).....	68
C.39. IE2 (SECONDARY INTERRUPT ENABLE – POVOLENÍ PŘERUŠENÍ 2).....	69
C.40. IEIP2 (SECONDARY INTERRUPT ENABLE AND PRIORITY – POVOLENÍ A PRIORITY PŘERUŠENÍ 2).....	69
C.41. INTVAL (USER TIME INTERVAL SELECT – UŽIVATELSKY NASTAVITELNÝ ČASOVAČ).....	69
C.42. IP (INTERRUPT PRIORITY – PRIORITY PŘERUŠENÍ).....	69
C.43. MIN (MINUTES TIME REGISTER – REGISTR MINUT).....	70
C.44. OF0H/M/(L) (PRIMARY ADC OFFSET CALIBRATION – POSUN PRIMÁRNÍHO A/D).....	70
C.45. OF1H/L (AUXILIARY ADC OFFSET CALIBRATION – POSUN POMOCNÉHO A/D).....	70
C.46. PLLCON (PLL CONTROL REGISTER – ŘÍDÍCÍ SLOVO PLL).....	70
C.47. PSMCON (POWER SUPPLY MONITOR – MONITOR NAPÁJENÍ).....	71
C.48. PWMCON (PWM CONTROL – ŘÍDÍCÍ SLOVO PWM).....	72
C.49. PWMxH/L (PWM DATA REGISTERS – DATA DO PWM PŘEVODNÍKU).....	72
C.50. SEC (SECONDS TIME REGISTER – REGISTR SEKUND).....	72
C.51. SPICON (SPI CONTROL – ŘÍDÍCÍ SLOVO SPI).....	73
C.52. SPIDAT (SPI DATA – DATA SPI KOMUNIKACE).....	73
C.53. TIMECON (TIC CONTROL – ŘÍDÍCÍ SLOVO TIC).....	74
C.54. WDCON (WATCHDOG TIMER – HLÍDACÍ ČASOVAČ).....	75
PŘÍLOHA D – INSTRUKČNÍ SADA 8051 (8052).....	76

Seznam symbolů a zkratek

A (Acc)	akumulátor – základní registr
A/D	analogově digitální převodník
ADC	Analog to Digital Converter – analogově digitální převodník
ADuC81x	ADuC812, ADuC814, ADuC816
ADuC83x	ADuC831, ADuC832, ADuC834, ADuC836
ADuC84x	ADuC841, ADuC842, ADuC843, ADuC845, ADuC847, ADuC848
ADuC8xx	kompletní řada mikrokontrolérů s označení začínající ADuC8
AGND	Analog Ground – analogová zem
AVDD	Analog Voltage Drain Drain – analogová kladné napájecí napětí
bit	binary digit – dvojková číslice
byte	slabika, je jednotka množství dat (osm bitů – osmiciferné binární číslo)
CPU	Central Processing Unit – centrální procesorová jednotka
D/A	digitálně analogový převodník
DAC	Digital to Analog Converter – digitálně analogový převodník
DGND	Digital Ground – digitální zem
DVDD	Digital Voltage Drain Drain – digitální kladné napájecí napětí
EAGLE	software pro návrh plošných spojů
EEPROM	Electrically Erasable Programmable Read-Only Memory – elektricky mazatelná paměť
Flash/EE	Období EEPROM
GND	Ground – zem
I2C	Inter-Integrated Circuit – multi-masterová sériová sběrnice
LED	Light-Emitting Diode – svítivá dioda
MQFP	Metric Quad Flat Package – SMD čtvercové pouzdro
OFF	vypnuto
ON	zapnuto
PC	Program Counter – čítač instrukcí (programový čítač)
PLL	Phase Locked Loop – smyčka fázového závěsu
PQFP	Plastic Quad Flat Package – SMD čtvercové pouzdro
PWM	Pulse Width Modulation – Pulsně šířková modulace
RAM	Random Access Memory – paměť s libovolným přístupem
Rn	registr R0-R7 aktuálně vybrané banky registrů
ROM	Read Only Memory – paměť s přístupem jen pro čtení
RxD	Receiver Data Input – přijímač dat
SFR	Special Function Registers – speciální funkční registry
SMD	surface mount device – součástka pro povrchovou montáž plošných spojů
SP	Stack Pointer – ukazovátka zásobníku
SPI	Serial Peripheral Interface – synchronní sériové komunikace
TIC	Time Interval Counter – časový intervalový čítač
TxD	Transmitter Data Output – vysílač dat
UART	Universal Asynchronous Receiver and Transmitter – asynchronní sériové rozhraní
USB	Universal Serial Bus – univerzální sériová sběrnice
VDD	Voltage Drain Drain – kladné napájecí napětí
C [F]	Kapacita [farad]
R [Ω]	Odpor [ohm]
L [H]	Indukčnost [henry]
U [V]	Napětí [volt]
I [A]	Proud [ampér]
t [$^{\circ}$ C]	Teplota [stupeň Celsia]

Úvod

Mikroprocesory způsobily ve vývoji elektroniky doslova revoluci stejně, jako v minulosti první tranzistor. Vývoj postupoval od nejjednodušších dvoubitových procesorů, až k dnešním 32, 64 a více bitovým. Na rozdíl od procesorů určených pro aplikace v osobních počítačích se dynamicky rozvíjí speciální kategorie mikroprocesorů, se snahou o sloučení více funkcí na jeden čip, určených pro aplikace v oblasti řízení, měření, předzpracování dat a jiné. Obvodům, které mají na čipu integrovanou paměť programu, paměť dat, hodinový oscilátor, sériový kanál, vstupně-výstupní obvody a mnohé jiné funkce, jako A/D, D/A převodník, Watch Dog aj. již přináleží název mikrokontrolérů, nebo jednočipové mikropočítače. Na počátku byla populární řada 8035 od firmy Intel, a později řada 8051. Díky vynikajícím vlastnostem byly, a ještě i nyní jsou tyto obvody používány ve velice širokém rozsahu. Mikrokontroléry produkují mnozí světoví výrobci elektronických součástek, přičemž někteří klonovali základní „INTELOvskou řadu“ a jiní postupovali nezávisle, vzhledem k původní řadě nekompatibilně. Tím je trh elektronických součástek obohacen o mnohé nové druhy mikrokontrolérů různých výrobců. U nás nejznámější byly mikrokontroléry vytvořené na základě řady 8035 a 8051 firmy INTEL. [1]

Cílem této diplomové práce je seznámení s produkty firmy  ANALOG DEVICES, konkrétně mikrokontroléry řady ADuC8xx, které jsou kompatibilní s mikrokontroléry řady INTEL 8051 a s jeho nástupcem INTEL 8052. Hlavním cílem je vytvoření ukázkových programů a ověření funkčnosti vývojového kitu vytvořeného v rámci mé bakalářské práce na kterou tato diplomová práce navazuje. Také bude provedeno měření proudu odebíraným mikrokontrolérem v normálním režimu a v režimu se sníženou spotřebou. Práce se bude také věnovat vytvoření ukázkového programu pro obsluhu teplotního senzoru TMP36, který bude připojený na interní A/D převodník. Dalším cílem diplomové práce je připojení a programová obsluha alfanumerického maticového displeje.

1. Architektura mikrokontrolérů ADuC8xx

Mikrokontroléry ADuC8xx byly první skutečnou integrací 12bitových až 24bitových přesných A/D převodníků a mikrokontroléru 8052. Blokové schémata jednotlivých typů jsou v příloze A. Vybrané části blokových schémat, které rozšiřují základní mikrokontrolér 8051 (8052) budou podrobněji popsány v dalších kapitolách.

Přehled všech vyráběných mikrokontrolérů ADuC8xx včetně základních parametrů uváděných výrobcem [15]:

Označen	MIPS	Flash [kB]	SRAM [B]	Vývodů	ADC			DAC výstupy (12-bit)	Další
					vstupů	rozišení [b]	vzorkovací rychlost		
ADuC812	1,3	8	256	34	8	12	200 kSPS	2	
ADuC814	1,3	8	256	17	6	12	247 kSPS	2	
ADuC816	1,3	8	256	34	4	16	105 SPS	1	
ADuC824	1	8	256	34	4	24	105 SPS	1	
ADuC831	1,3	62	2 304	34	8	12	247 kSPS	2	PWM
ADuC832	1,3	62	2 304	34	8	12	247 kSPS	2	PWM
ADuC834	1	62	2 304	34	4	24	105 SPS	1	PWM
ADuC836	1	62	2 304	34	4	16	105 SPS	1	PWM
ADuC841	20	62	2 304	34	8	12	400 kSPS	2	PWM
ADuC842	16	62	2 304	34	8	12	400 kSPS	2	PWM
ADuC843	16	62	2 304	34	8	12	400 kSPS	0	PWM
ADuC845	12	62	2 304	34	10	24	1,37 kSPS	1	PWM
ADuC847	12	62	2 304	34	10	24	1,37 kSPS	1	PWM

Tab. 1.1 Seznam mikrokontrolérů ADuC8xx a jejich základní parametry [15]

1.1. Organizace paměti

Všechny mikrokontroléry ADuC8xx mají oddělený adresový prostor programu a adresový prostor dat. Tato koncepce je označovaná jako Harwardské uspořádání.

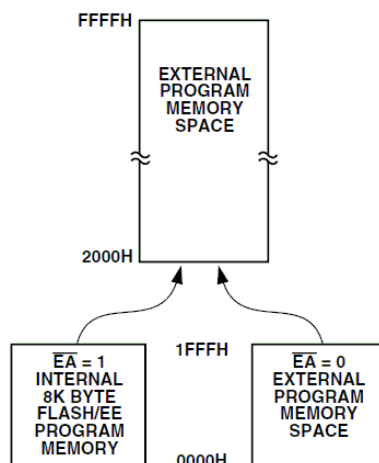
1.1.1. Paměť programu

Paměť programu je v provedení Flash EEPROM. Vnitřní adresový prostor paměti programu je 8 kB nebo 62 kB podle typu mikrokontroléru (viz Tab. 1.1). Tento adresový prostor paměti je možno rozšířit na maximální velikost 64 kB nebo úplně nahradit externí pamětí (viz Obr. 1.1). Typ použité paměti se volí vstupem \overline{EA} , kde 0 znamená použití externí paměti nebo 1 znamená interní paměť, což se využívá častěji. Na nulté stránce programové paměti jsou rezervované vektorové adresy jednotlivých přerušení.

Vlastní běh programu je adresován 16bitovým čítačem instrukcí PC.

Source	Vector Address
IE0	0003h
TF0	000Bh
IE1	0013h
TF1	001Bh
RI + TI	0023h
TF2 + EXF2	002Bh
RDY0/RDY1 (ADC)	0033h
ISPI	003Bh
PSMI	0043h
TII	0053h
WDS	005Bh

Tab. 1.2 Vektorové adresy přerušení [5], [6], [7], [8], [9], [10], [11], [12], [13], [14]



Obr. 1.1 Architektura paměti programu (8 kB interní paměť) [5], [6], [7], [8], [9], [10], [11], [12], [13], [14]

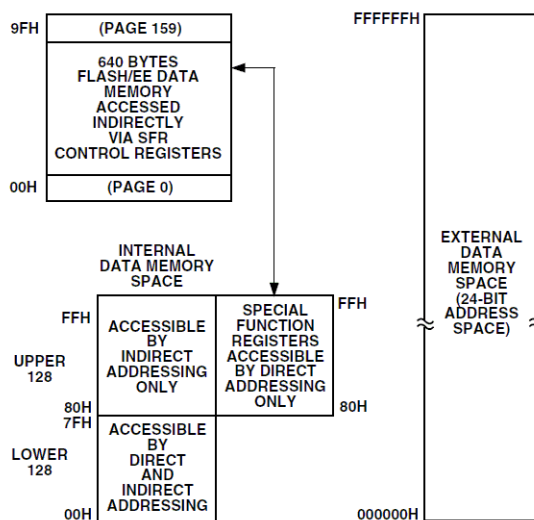
1.1.2. Paměť dat

Paměť dat mikrokontrolérů řady ADuC8xx je možno rozdělit na interní a externí. Interní paměť dat typu RAM má rozsah 256 bytů. Tento paměťový prostor je rozdělen na tři části:

LOWER – dolních 128 bytů,

UPPER – horních 128 bytů,

SPECIAL FUNCTION REGISTERS (SFR) – oblast speciálních registrů



Obr. 1.2 Architektura paměti dat [5], [6], [7], [8], [9], [10], [11], [12], [13], [14]

Popis paměťového prostoru UPPER a LOWER a jejich adresování není součástí této práce, bližší informace jsou k dispozici v literatuře např. [1].

Paměť prostoru SFR je umístěna na adresách 80h až FFh. Organizace a způsob adresace speciálních registrů je patrný z tabulek v příloze B. Všechny registry končící na 0h nebo 8h je možné adresovat bitově. Registry převzaté přímo z architektury řady 8051 a 8052 mají pevně určenou pozici. Popis těchto registrů a jejich funkci je možno taktéž najít v literatuře. Registry 8051 (8052) byly u rodiny mikrokontrolérů ADuC8xx doplněny o další konfigurační registry obsluhující integrované převodníky a další nastavení a funkce. Jejich význam a popis je uveden v příloze C.

1.2. Integrované uživatelské funkční bloky

1.2.1. Paměť Flash/EE

Mikrokontroléry ADuC8xx mají na čipu implementovanou Flash/EE paměť, kde v jedné části je uchován vlastní program a do druhé části je možno uložit data. Velikost datové části je 640 byte nebo 4096 byte podle typu mikrokontroléru (viz příloha C.25 EDARL/(H)). Do datové části není přímý přístup, ale pomocí šesti (sedmi) speciálních registrů. Přístup do datové Flash/EE se volí pomocí řídicího slova ECON, adresy EDARL/(H), datových registrů EDATA1-4. Podrobněji jsou tyto registry vysvětleny v příloze C.

1.2.2. A/D převodník

Dalším blokem na čipu je převodník(y) analogového signálu na digitální signál. U většiny typů mikrokontrolérů ADuC8xx je osazen jeden 12bitový až 24bitový převodník a multiplexer přepínání analogových vstupů, vnitřního teplotního čidla a v některých variantách zpětné vazby z D/A převodníků a referenčního napětí. U mikrokontrolérů ADuC816, ADuC824, ADuC834, ADuC836 a ADuC845 je primární převodník doplněný o pomocný převodník. Vlastní konfigurace, rozlišení převodníků a počtu analogových vstupů je patrné z blokových schémat v příloze A a Tab. 1.1. Chování A/D převodníků lze řídit pomocí speciálních registrů ADCxCON, ADCCONx, ADCGAINH/L, ADCMODE, ADCOFSH/L, ADCSTAT, GN0H/M/(L), GN1H/L, OF0H/M/(L) a OF1H/L. Výstupní hodnoty lze potom získat ve speciálních registrech ADC0H/M/(L) ADC1H/L nebo ADCDATAH/L. Podrobněji jsou, všechny výše uvedené registry, vysvětleny v příloze C.

1.2.3. D/A převodník

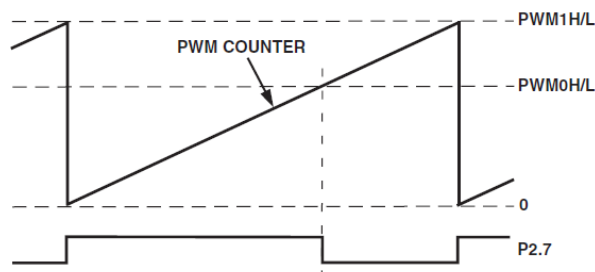
U všech typů mikrokontrolérů ADuC8xx vyjma ADuC843 je na čipu integrován jeden nebo dva 12bitové převodníky z digitálního signálu na analogový. Počet převodníků je možné najít v Tab. 1.1. Chování D/A převodníku je možné nastavit ve speciálním registru DACCON a vlastní data se zapisují do registrů DACH/L nebo DACxH/L. Podrobněji jsou, všechny výše uvedené registry, vysvětleny v příloze C.

1.2.4. Pulsně šířkový modulátor PWM

U mikrokontrolérů řady ADuC83x a ADuC84x je na čipu implementován pulsně šířkový modulátor u kterého lze řídit režim pomocí speciálního registru PWMCON zadaných hodnot v registrech PWMxH/L (viz Příloha C). PWM modulátor je možné provozovat v několika režimech.

1.2.4.1. PWM režim 1

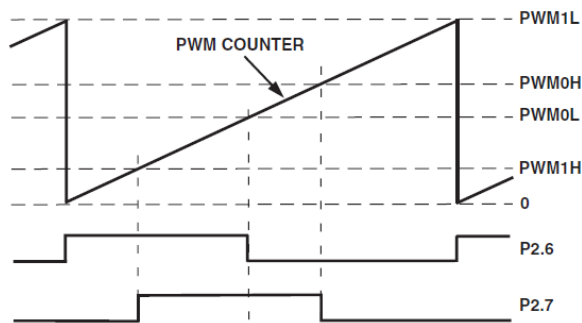
V režimu 1 je možno nastavit jak délku pulsu, tak dobu cyklu jako 16bitové číslo. Číslo uložené v registrech PWM0H/L určuje délku pulsu a PWM1H/L určuje celkovou dobu cyklu. V tomto režimu je možné provozovat jen jeden výstup PWM0.



Obr. 1.3 PWM režim 1 [9], [10], [11], [12], [13], [14]

1.2.4.2. PWM režim 2

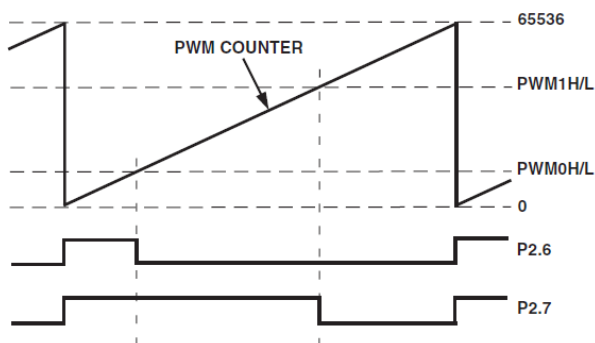
V režimu 2 je možné provozovat oba výstupy PWM, ale hodnoty je možno nastavit jen 8bitově. Číslo uložené v registru PWM1L určuje celkovou dobu cyklu, PWM0L určuje délku pulsu PWM0, PWM1H určuje vzájemný posun startů pulsu PWM0 a PWM1. Délka pulsu PWM1 je určena rozdílem PWM0H a PWM1H.



Obr. 1.4 PWM režim 2 [9], [10], [11], [12], [13], [14]

1.2.4.3. PWM režim 3

V režimu 3 je možné provozovat oba výstupy PWM a hodnoty je možno nastavit 16bitově. Celková doba cyklu je pevně daná na 65536. Číslo uložené v registru PWM0H/L určuje délku pulsu PWM0 a PWM1H/L určuje délku pulsu PWM1. Oba výstupy začínají synchronně.

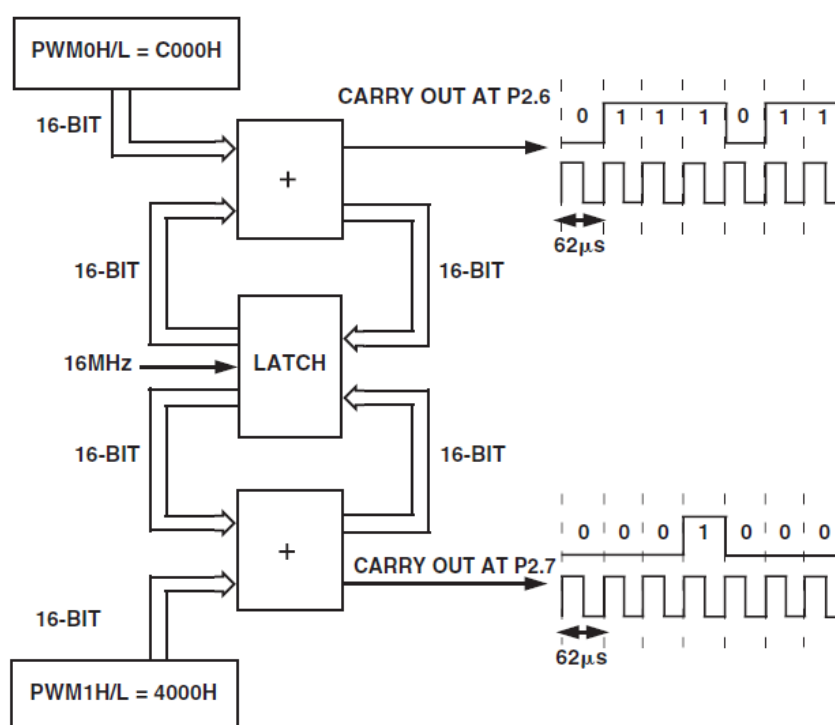


Obr. 1.5 PWM režim 3 [9], [10], [11], [12], [13], [14]

1.2.4.4. PWM režim 4

V režimu 4 PWM funguje jako $\Sigma - \Delta$ DA převodník. Pro tento režim je doporučená modulační frekvence f_{VCO} nebo f_{OSC} (viz příloha C.48 PWMCON). V tomto režimu jsou, každý hodinový puls (62 ns v případě taktovací frekvence 16 MHz), výstupy PWM0 a PWM1 aktualizovány. Během všech 65536 cyklů (16 bitů PWM) jsou rovnoměrně rozmístěny logické 1 a 0, kdy počet logických 1 je PWM0H/L a počet logických 0 je 65536 - PWM0H/L pro výstup PWM0. Podobně pro PWM1 PWM1H/L = počet log. 1 a 65536 - PWM1H/L = počet log. 0.

Například, pokud hodnota PWM1H/L bude 4010h (mírně nad jednu čtvrtinu) pak výstup PWM1 bude v logické 0 tři takty a v logické 1 jeden takt. Protože hodnota je mírně nad jednu čtvrtinu, bude se výstup kompenzovat několikrát za 65536 cyklů jen dvěma logickými 0.

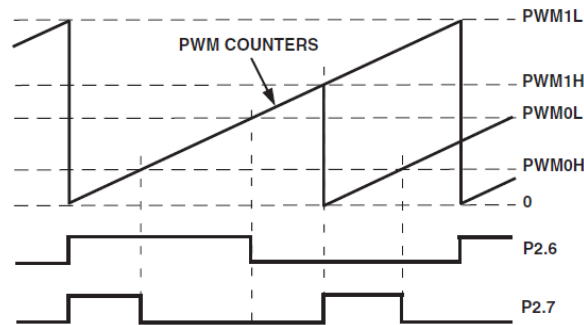


Obr. 1.6 PWM režim 4 [9], [10], [11], [12], [13], [14]

Pokud bude potřeba použít hodnoty s nižším rozlišením (počet bitů je menší než 16), musí být hodna v nejvyšších bitech doplněna log. 0 v nejnižších bitech. Je-li například požadovaná hodnota 8bitové číslo, je nutné zapsat požadovanou hodnotu do PWMxH a do PWMxL je nutné zapsat 00h.

1.2.4.5. PWM režim 5

V režimu 5 je možné provozovat oba výstupy PWM, ale hodnoty je možno nastavit jen 8bitově. Číslo uložené v registru PWM1L určuje celkovou dobu cyklu PWM0, PWM0L určuje délku pulsu PWM0. Číslo uložené v registru PWM1H určuje celkovou dobu cyklu PWM1, PWM0H určuje délku pulsu PWM1. Oba výstupy PWM mají stejný zdroj hodin a dělič.

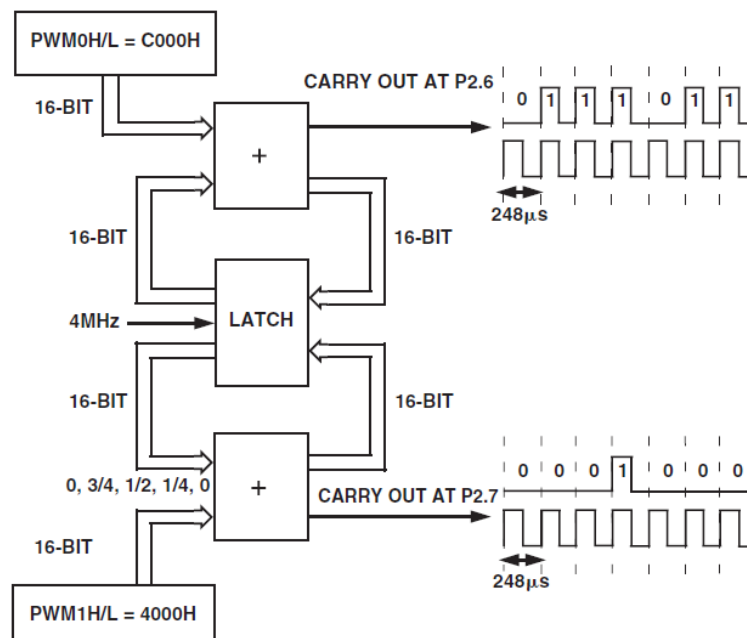


Obr. 1.7 PWM režim 5 [9], [10], [11], [12], [13], [14]

1.2.4.6. PWM režim 6

V režimu 6 PWM funguje jako $\Sigma - \Delta$ DA převodník. Pro tento režim je doporučená modulační frekvence f_{VCO} nebo f_{OSC} (viz. příloha C.48 PWMCON) s děličem PWM hodin roven 4. V tomto režimu jsou, každý hodinový puls (248 ns v případě taktovací frekvence 16 MHz), výstupy PWM0 a PWM1 aktualizovány. Během všech 65536 cyklů (16 bitů PWM) jsou rovnoměrně rozmístěny logické 1 a 0, kdy počet logických 1 je PWM0H/L a počet logických 0 je 65536 - PWM0H/L pro výstup PWM0. Podobně pro PWM1 PWM1H/L = počet log. 1 a 65536 - PWM1H/L = počet log. 0. Doba trvání logické 1 je jen polovinu taktu a druhá polovina doplněna logickou 0.

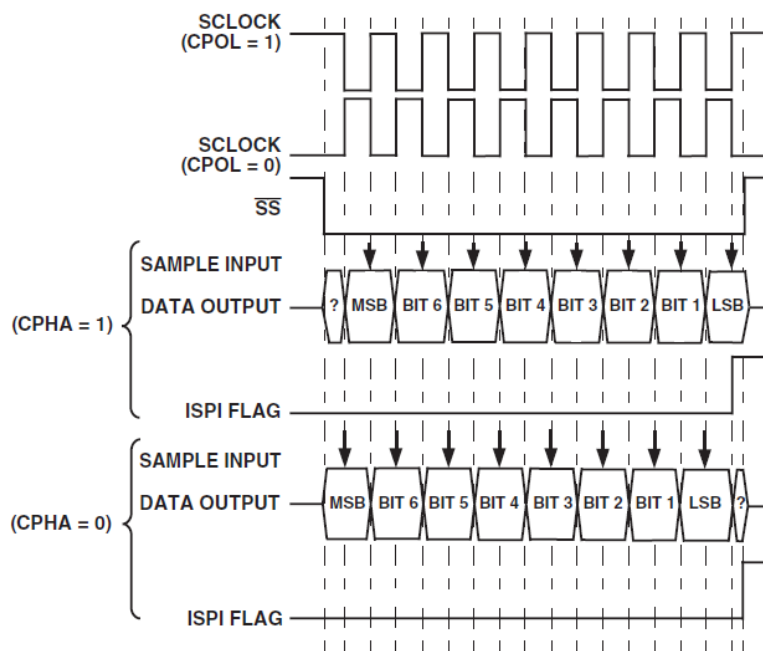
Například, pokud hodnota PWM1H/L bude 4010h (mírně nad jednu čtvrtinu) pak výstup PWM1 bude v logické 0 tři takty (3×248 ns), v logické 1 půl taktu (124 ns) a v logické 0 půl taktu (124 ns). Protože hodnota je mírně nad jednu čtvrtinu, bude se výstup kompenzovat několikrát za 65536 cyklů jen dvěma logickými 0.



Obr. 1.8 PWM režim 6 [9], [10], [11], [12], [13], [14]

1.2.5. SPI komunikace

ADuC8xx integruje kompletní hardwarovou Serial Peripheral Interface (SPI) na čipu. SPI je průmyslovým standardem synchronní sériové komunikace, která umožňuje synchronně vysílat a přijímat současně osm bitů dat (plně duplexní). Pozor SPI komunikace sdílí vstup-výstupní vývody s I²C komunikací, proto může uživatel povolit jen jeden druh komunikace v dané době (bit SPE v řídicím slově SPICON). Chování SPI je možno nastavit ve SPICON a data se odesílají (přijímají) pomocí registru SPIDAT. Význam bitů SPICON je podrobněji vysvětlen v příloze C.



Obr. 1.9 SPI časování, všechny druhy provozu [5], [6], [7], [8], [9], [10], [11], [12], [13], [14]

1.2.5.1. SPI – master mode

V režimu master, je vývod SCLOCK nastaven jako výstup a generuje osm impulzů hodin, kdykoliv se zapíše data do registru SPIDAT. Data z tohoto registru jsou odesílána pomocí MOSI. Zároveň jsou přijímána data přicházející do MISO. Po osmi pulzech SCLOCK jsou přijatá data uložena do registru SPIDAT, je zavoláno přerušení a nastaveno ISPI. Vývod \overline{SS} není v master modu používán, pokud je pro provoz slave zařízení potřebný, musí být obslužen programově.

1.2.5.2. SPI – slave mode

V režimu slave, je vývod SCLOCK nastaven jako vstup. Vstup \overline{SS} musí být v logické 0. Pokud přicházejí hodinové pulsy na SCLOCK jsou přijímána data přes MISO a zároveň je obsah SPIDAT odeslán pomocí MOSI. Po osmi pulzech SCLOCK jsou přijatá data uložena do registru SPIDAT a zavoláno přerušení a nastaveno ISPI.

1.2.6. I²C kompatibilní rozhraní

ADuC8xx podporuje plně, licencovanou I²C sériovou komunikaci (Licence Philips). I²C rozhraní je implementováno s úplným hardwarovým řízením ve slave režimu a softwarovým řízením v master režimu. Vstup-výstup SDATA je použit pro vlastní přenos a SCLOCK jsou synchronizační hodiny. Tyto dva vývody jsou sdíleny se sériovou komunikací SPI. Vzhledem k tomu může být povolen pouze jeden druh komunikace (viz kap. 1.2.5 SPI komunikace). Vlastní komunikaci je možno nastavit v registru I2CCON, adresa masteru (slave) je v registru I2CADD a datový registr je I2CDAT. Význam bitů I2CCON je podrobněji vysvětlen v příloze C.

1.2.7. Monitor napájení

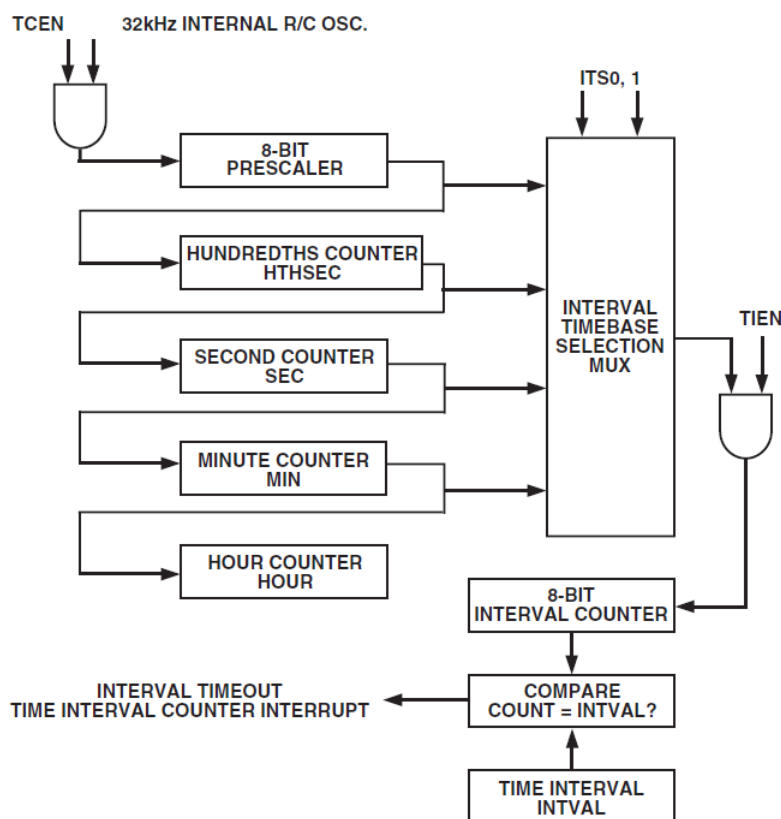
Pomocí této funkce je možno sledovat napětí DVDD případně AVDD, které se porovná s nastavenou úrovní, v případě že dojde k podkročení požadované hodnoty je nastaven příznak případně je zavoláno příslušné přerušení. Tato funkce umožňuje uživateli uložit pracovní data, aniž dojde ke ztrátě způsobené nízkým napájecím napětím. Dále zajišťuje, že normální spuštění programu nebude pokračovat, dokud nebude prokázána bezpečná úroveň napětí. Monitor je možno ovládat pomocí PSMCON (viz Příloha C).

1.2.8. Hlídač běhu (Watchdog timer)

Účelem watchdog (hlídače běhu) je provést reset zařízení nebo provést přerušení v přiměřeném času, pokud u ADuC8xx dojde k chybnému stavu, buď kvůli programové chybě, nebo elektrickému rušení. Watchdog funkci lze ovládat pomocí registru WDCON (viz Příloha C). Vlastní časovač je 16bitový čítač, který je taktovaný na 32 kHz vnitřním R/C oscilátorem nebo externím krystalem (podle typu a nastavení). Watchdog časový limit lze nastavit pomocí bitů PRE0-3 WDCON.

1.2.9. Časový čítač (TIC)

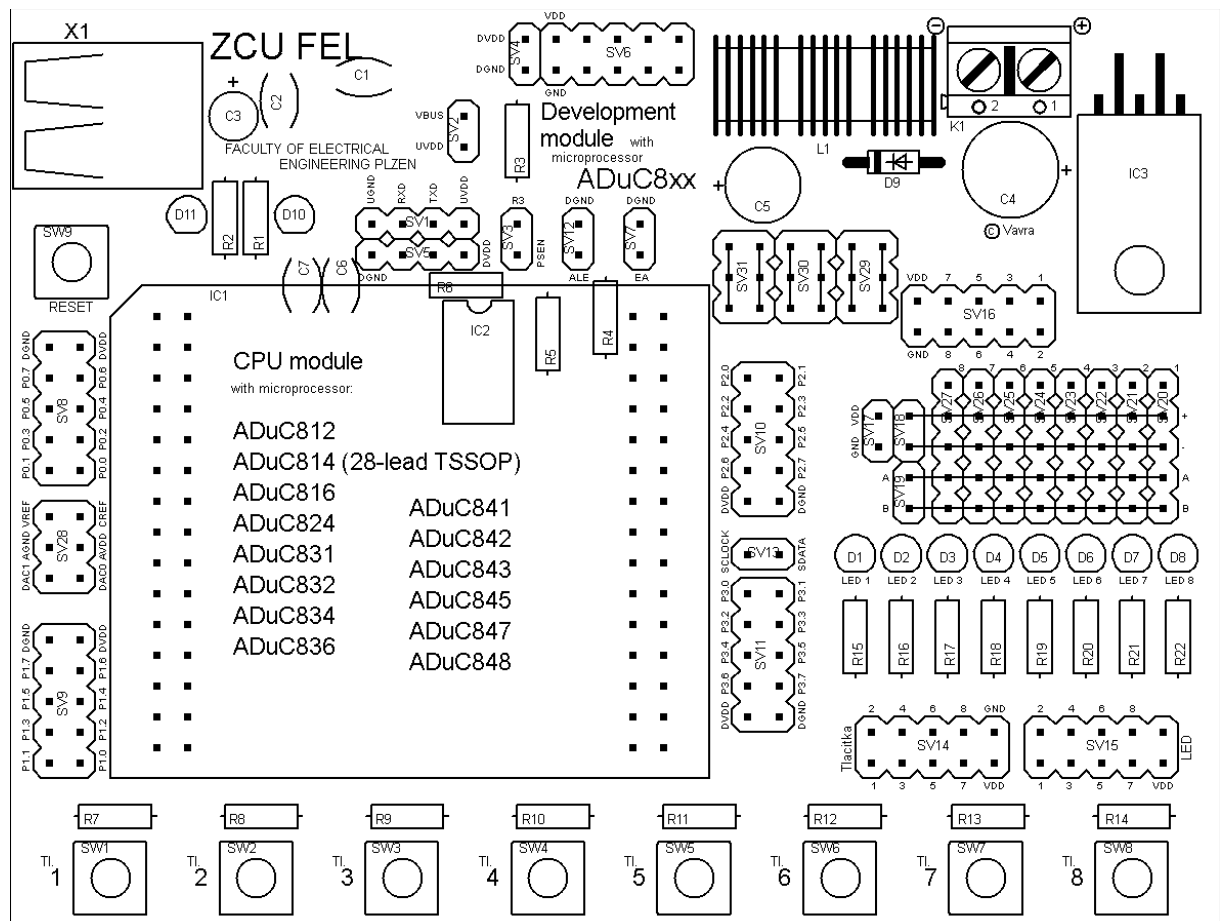
Časový intervalový čítač (TIC) je na čipu k dispozici pro počítání delších časových intervalů, než jsou schopni standardní časovače kompatibilní s 8051. TIC je schopen čítat v intervalech od 1/128 sekundy do 255 hodin. Dále je tento čítač časován vnitřním R/C oscilátorem nebo externím krystalem a má schopnost zůstat aktivní v power-down módu. To má výhodu u bateriově napájených aplikací, kdy může být systém pravidelně buzen. Nastavení čítače se provádí v registru TIMECON a požadovaný čas v registru INTVAL. Časová základna pro svoji funkci potřebuje ještě čtyři registry (HSECS, SEC, MIN, HOUR), které je možné použít jako časovou základnu reálného času. Podrobněji jsou, všechny výše uvedené registry, vysvětleny v příloze C.



Obr. 1.10 Bloková schéma časového intervalového čítače [5], [6], [7], [8], [9], [10], [11], [12], [13], [14]


2. Vývojový modul

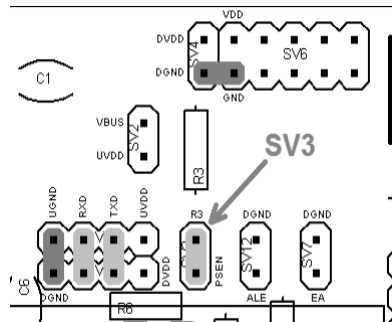
Při návrhu vývojového modulu jsem se nechal inspirovat řešením výrobce Atmel STK 500 [18], kde základní myšlenkou je deska, na které jsou osazeny základní periferie, napájení, komunikační (programovací) rozhraní. Za výhodu považuji možnost vložení různých typů procesorů. Pro návrh plošných spojů jsem zvolil software EAGLE Light Edition, který je volně k dispozici na internetových stránkách výrobce. U této verze software je limitována velikost plochy desky na 100 × 80 mm (4 × 3.2 palce). Na základě všech těchto skutečností jsem se rozhodl navrhnout vývojový modul rozdělený na několik částí. První částí je CPU modul, který je osazen mikrokontrolérem ADuC8xx v 52vývodovém pouzdru PQFP (v některé odborné literatuře označováno MQFP). Druhou částí je vlastní vývojový kit, obsahující zdroj 5VDC, komunikační převodník, testovací tlačítka a signalizační LED. Podrobněji byla stavba popsána v méj bakalářské práci, na kterou tato práce navazuje.



Obr. 2.1: Rozmístění součástek a popis přední strany vývojového kitu ADuC8xx

2.1. Nahrávání programu do mikrokontroléru

Pro nahrávání programu ve formě strojového kódu do mikrokontroléru je určena aplikace Windows Serial Downloader [20] přímo od výrobce rodiny mikrokontrolérů ADuC8xx – firmy . Strojový kód se nahrává přes rozhraní USB – UART. Pro vstup do režimu umožňujícího nahrávání je potřeba, aby vývod PSEN byl při startu ADuC8xx uzemněn přes rezistor R3 (1 kΩ) pomocí jumperu SV3. Po nahrání programu je potřeba pro normální chod mikrokontroléru naopak tento jumper rozpojit.



Obr. 2.3 Doporučené propojení sériové komunikace a PSEN



Obr. 2.4 Okno aplikace Windows Serial Downloader

2.2. Externí periferie a ukázkový program

Pro pochopení dalších částí textu je předpokladem alespoň základní znalost assembleru a architektury mikrokontrolérů řady 8051. Pro sjednocení terminologie uvedu některé pojmy (čerpáno z [1], [2], [3]): Instrukční cyklus – množina dob nezbytných k vykonání jedné instrukce. Pro různé instrukce má jeden instrukční cyklus různý počet strojových dob (viz příloha D). U mikrokontrolérů řady ADuC8xx se jedna strojová doba rovná jednomu taktu krystalu.

Strojová instrukce – dále jen instrukce, je kódový příkaz k vykonání jisté strojové operace. Úplný soubor instrukcí je u každého procesoru nejnižší programovací úrovní dostupnou uživateli. Kompletní soubor strojových instrukcí mikrokontrolérů řady 8051 (instrukční sada) je uveden v příloze D.


Assembler – doslova „sestavovací“ jazyk, oficiální český název je jazyk symbolických adres. Pro tvorbu programů ve strojové podobě je nutné programovat v jakési vyšší podobě strojových instrukcí, každé strojové instrukci přísluší určitý mnemokód – zkratka vystihující funkci instrukce. Z těchto zkratk se vytváří program, který se pomocí kompilátoru a linkeru přeloží tento zdrojový kód do definitivní podoby – do strojového kódu. Kompilátor je počítačový program, který transformuje zdrojový kód do objektového souboru. Linker následně tento nebo více objektových souborů spojí do jednoho souboru obsahující výsledný strojový kód. Kompilátor a linker se souhrnně nazývají překladač. Pro práci překladače se používají další příkazy, které se v překladu neobjevují (nemají strojovou interpretaci), ale usnadňují překlad nebo psaní programu v assembleru.

Makroinstrukce – předem definovaná posloupnost instrukcí. Je to vlastně pokyn pro překladač, aby do cílového programu zařadil žádoucí skupinu. Šetří se tím čas programátora – jednou vytvořená posloupnost instrukcí se celá vloží místo makroinstrukce, v žádném případě se cílový program nezkrátí.

Pseudoinstrukce – příkaz pro překladač. Může sloužit k nastavení adresy, od které se program bude ukládat a podobně.

V následujících kapitolách bude popsáno několik příkladů ukázkových programů a testů funkčních celků implementovaných na čipu mikrokontroléru. Všechny programy byly testovány na vývojovém kitu osazený mikrokontrolérem ADuC841 s krystalem s taktovací frekvencí 11,0592 MHz.

2.2.1. Režim se sníženou spotřebou

U aplikací, kde není vyžadována soustavná činnost a jsou kladeny přísné požadavky na spotřebu energie například u napájení z baterií, je obvykle nutné u procesorů využívat režimů pro sníženou spotřebou. Provedl jsem měření spotřeby mikrokontroléru ADuC841, který je osazen na prototypu vývojového modulu. Pro toto měření bylo použito originální ukázkový program od výrobce, firmy  [19]

Upravený, okomentovaný programový výpis:

```

$MOD841                ; definice 8052 a ADuC841
LED EQU P3.4           ; inicializace proměnné - výstup s připojenou LED
ORG 0000h
LJMP BEGIN             ; skok na začátek programu
                       ; adresace přerušení
ORG 0003h              ; interrupt IE0 (vnější přerušení 0)
RETI
ORG 000Bh              ; interrupt TF0 (přerušení od časovače 0)
ORG 0013h              ; interrupt IE1 (vnější přerušení 1)
ORG 001Bh              ; interrupt TF1 (přerušení od časovače 1)
ORG 0023h              ; interrupt Serial (přerušení od sériového rozhraní)
ORG 0053h              ; interrupt TIC (přerušení od časového intervalového čítače)
RETI
ORG 0060h

BEGIN:
MOV IEIP2,#0A4h        ; povolení přerušení od časového intervalového čítače
SETB IT0               ; INT0 přerušení od náběžné hrany
SETB EX0               ; povolení vnějšího přerušení INTO
SETB EA                ; povolení všech přerušení
MOV INTVAL,#14h        ; nastavení INTVAL na 20 časových jednotek (viz TIMECON)
BLINK: MOV R0,#50       ; 50 cyklů – 25 bliknutí LED (100ms vyp / 100ms zap)
LOOP: MOV A,#010        ; nastavení počtu cyklů čekání 10 =>čekání 100ms
CALL DELAY             ; volání podprogramu čekání
CPL LED                ; změna stavu výstupu (LED)
DJNZ R0, LOOP          ; R0=R0-1 a skok na LOOP, pokud R0 není 0
MOV TIMECON,#13h       ; nastavení TIMECON (čítání v sekundách, start čítání)
SETB LED               ; vypnutí LED před přepnutím do režimu se sníženou spotřebou
MOV PCON,#22h          ; zapnutí režimu se sníženou spotřebou
MOV TIMECON,#12h       ; zakázání TCEN pro zastavení časovače a nastavení na 0
JMP BLINK              ; skok na začátek hlavní smyčky
DELAY:                 ; doba cyklu DLY0 na základě 11,0592MHz taktu jádra
                       ; (2 + 144 × (2 + 255 × 3 + 3) + 3) / 11 059 200 = 0,01s
MOV R5,A               ; uložení počtu cyklů čekání z akumulátoru do R5
DLY0: MOV R6,#090H      ; nastavení zpoždění – cyklus 0, R6 = 144 (2 strojové doby)
DLY1: MOV R7,#0FFH      ; nastavení zpoždění – cyklus 1, R7 = 255 (2 strojové doby)
DJNZ R7,$              ; R7=R7-1 pokud R7 není 0 (3 strojové doby)
DJNZ R6,DLY1           ; R6=R6-1 a skok na DLY1, pokud R6 není 0 (3 strojové doby)
DJNZ R5,DLY0           ; R5=R5-1 a skok na DLY0, pokud R5 není 0 (3 strojové doby)
RET                    ; návrat z podprogramu
END

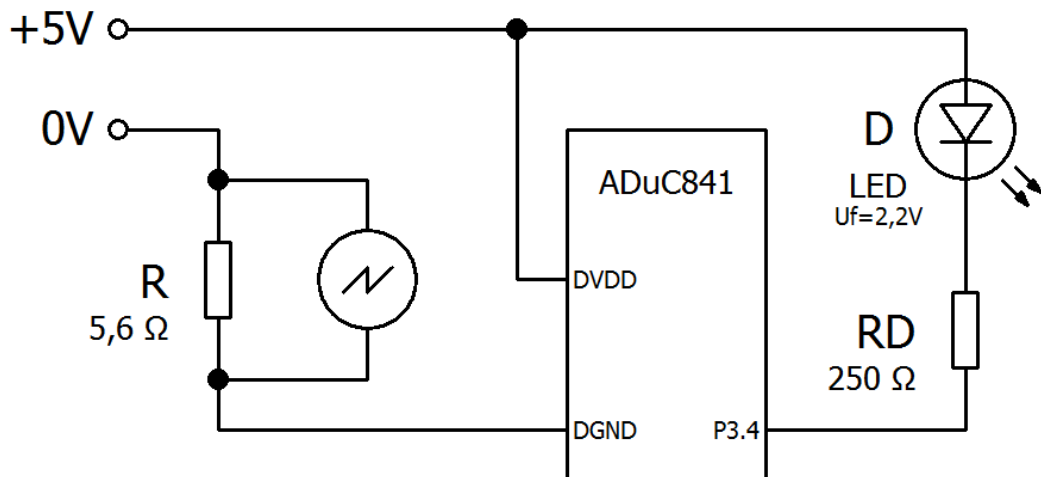
```


Při měření spotřeby mikrokontroléru byl zjišťován úbytek napětí pomocí osciloskopu Agilent HP MSO6054A na rezistoru v záporné napájecí větvi mikrokontroléru. Schéma zapojení měření spotřeby je na Obr. 2.5. Rezistor v záporné větvi jsem volil tak, aby výsledné napájecí napětí mikrokontroléru bylo minimálně 4,75 V, ale zároveň aby úbytek napětí byl měřitelný s dostatečnou přesností. Zvolil jsem úbytek napájecího napětí 0,2 V. Maximální předpokládaný proud je součet proudů odebíraných mikrokontrolérem a sepnutou svítivou diodou. Katalogová hodnota proudu odebíraného mikrokontrolérem pro danou taktovací frekvenci a napájecí napětí 5 V není uvedena, na základě hodnot pro ostatní frekvence jsem ho odhadl na 27 mA. Protože svítivá dioda je spínána přímo výstupem mikrokontroléru přes ochranný rezistor je proud protékající přes svítivou diodu omezen velikostí ochranného rezistoru. Napětí na rezistoru vypočteme odečtením úbytku napětí na svítivé diodě, spínacím prvku mikrokontroléru a měřicím rezistoru od napájecího napětí 5 V. Katalogový údaj U_f svítivé diody je 2,2 V a napětí na výstupu z mikrokontroléru při log. 0 je 0,4 V.

$$I_{RD} = \frac{U - U_{fLED} - U_{LOW} - U_R}{R_D} = \frac{5,0 \text{ V} - 2,2 \text{ V} - 0,4 \text{ V} - 0,2 \text{ V}}{250 \Omega} = 0,0088 \text{ A} = 8,8 \text{ mA}$$

$$I_{MAX} = I_{RD} + I_{CPU} = 8,8 \text{ mA} + 27 \text{ mA} = 35,8 \text{ mA}$$

$$R = \frac{U_{RMAX}}{I_{MAX}} = \frac{0,2 \text{ V}}{0,0358 \text{ A}} = 5,59 \Omega$$



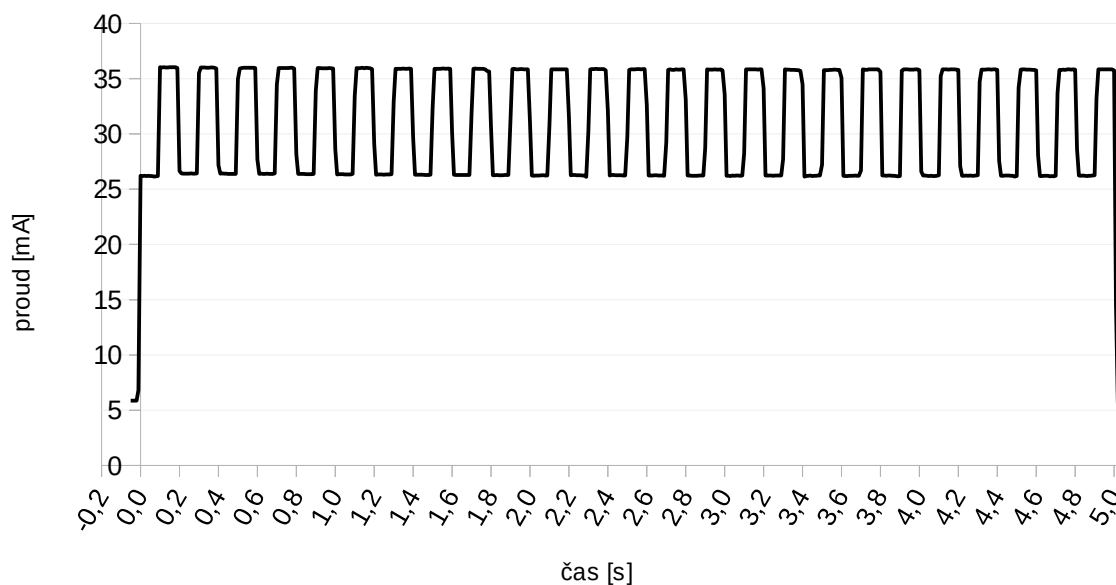
Obr. 2.5 Schéma zapojení měření spotřeby mikrokontroléru

Naměřená data z osciloskopu jsem zpracoval pomocí tabulkového procesoru.

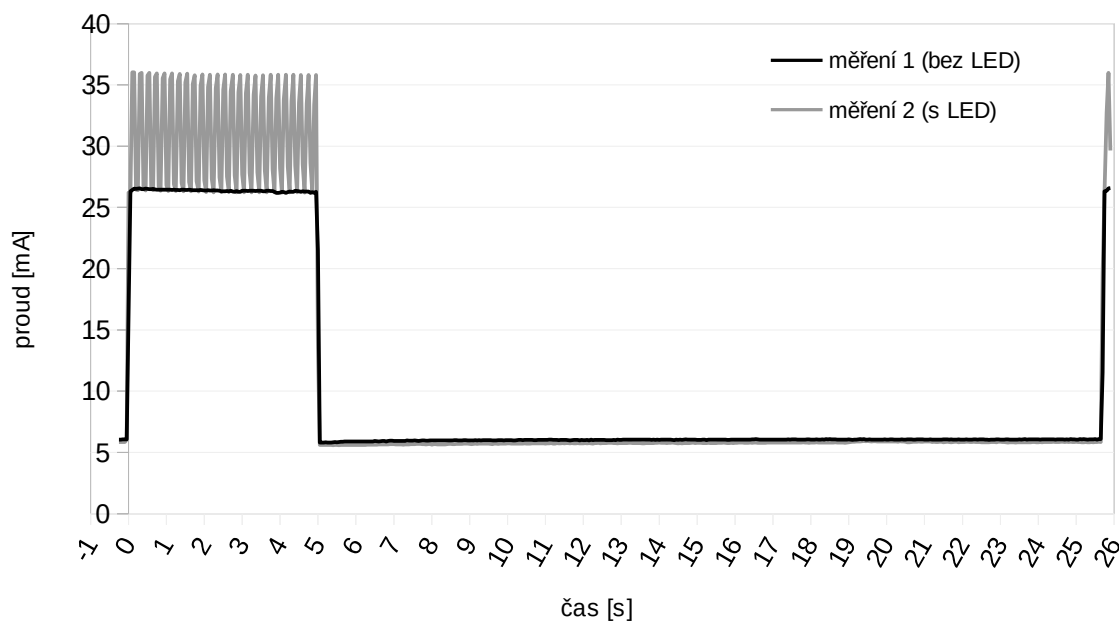
$$I = \frac{U_R}{R} = \frac{U_R}{5,6 \Omega}$$

Průběhy proudu v jednotlivých režimech jsou vidět na výsledných grafech (Obr. 2.6 a Obr. 2.7)

V normálním režimu mikrokontroléru jsem naměřil proud 26,5 mA, v normálním režimu se sepnutou svítivou diodou byl celkový proud 35,7 mA. Proud sepnuté svítivé diody je 9,2 mA, to odpovídá reálnému naměřenému napájecímu napětí $U = 5,19$ V a ochrannému odporu $R_D = 237 \Omega$.



Obr. 2.6: Graf průběhu proudu mikrokontroléru v normálním režimu včetně spínání LED



Obr. 2.7: Graf průběhu proudu celého cyklu mikrokontroléru bez/s LED

Pro účely zjištění odebírané energie v režimu se sníženou spotřebou je podstatnější průběh proudu samotného mikrokontroléru, tzn. s odpojenou svítivou diodou. V režimu se sníženou spotřebou jsem naměřil proud 6,1 mA. Celkový čas cyklu průběhu programu je 25,7 s a z toho 5 s je mikrokontrolér v normálním režimu. Průměrný proud odebíraný mikrokontrolérem jsem vypočetl integrací z naměřených hodnot:

$$I_p = \frac{\int_0^T I(t) dt}{T} = \frac{\int_0^{T_1} I_1(t) dt + \int_{T_1}^T I_2(t) dt}{T} = \frac{(I_1 \cdot T_1 - I_1 \cdot 0) + (I_2 \cdot T - I_2 \cdot T_1)}{T}$$

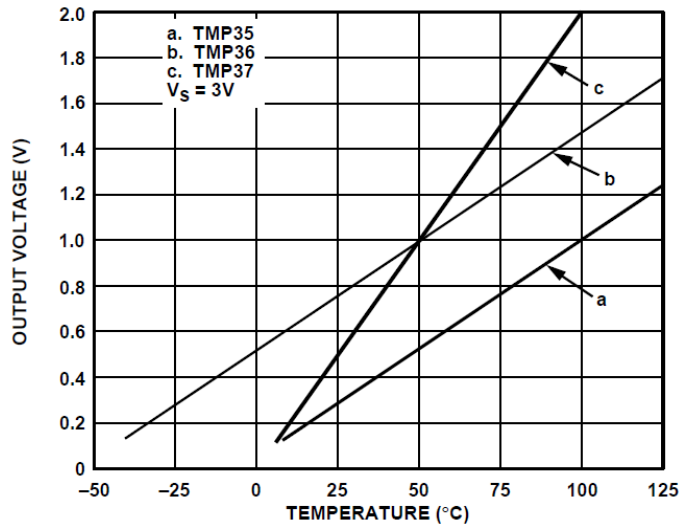
Kde $T=25,7\text{s}$, $T_1=5\text{s}$, $I_1=26,5\text{mA}$, $I_2=6,1\text{mA}$.

$$I_p = \frac{\int_0^5 26,5 dt + \int_5^{25,7} 6,1 dt}{25,7} = \frac{(26,5 \cdot 5 - 26,5 \cdot 0) + (6,1 \cdot 25,7 - 6,1 \cdot 5)}{25,7} = 10,1 \text{ mA}$$

Pokud se bude tato aplikace napájet akumulátorem s kapacitou 1Ah při použití režimu se sníženou spotřebou bude možné provozovat mikrokontrolér 4 dny (tj. 99 hod. 19 min.). Naopak bez aktivace režimu se sníženou spotřebou je možné provozovat mikrokontrolér jen 1,5 dne (tj. 37 hod. 44 min.).

2.2.2. Snímač teploty s analogovým výstupem (TMP36)

Měření teploty pomocí snímače TMP36 je možné v rozsahu od $-40\text{ }^{\circ}\text{C}$ do $+125\text{ }^{\circ}\text{C}$. Tento třívodičový teplotní snímač v pouzdře TO-92 je možné napájet napětím od 2,7 V do 5,5 V a jeho výstupní napětí je od 0 mV do 1750 mV (viz křivka *b* na Obr. 2.8).



Obr. 2.8: Závislost výstupního napětí snímače na teplotě [16]

Teplotu zde je možné vyjádřit:

$$t = \frac{U - 500}{10} \text{ [}^{\circ}\text{C]} - \text{kde } U \text{ je v milivoltech}$$

Pokud se signál ze snímače bude zpracovávat interním A/D převodníkem, který má rozsah 0 až 2,5 V na 4096 dílků, je možné teplotu vypočítat:

$$t = \frac{(AD0 \cdot 5 - 4096) \cdot 25}{2048} \text{ [}^{\circ}\text{C]} - \text{kde } AD0 \text{ je hodnota z ADC kanálu 0}$$

Protože je v mikrokontroléru implementována jen základní bytová matematika, je dobré volit výpočet tak, aby výpočet byl co nejjednodušší a zároveň aby nedošlo k velkým chybám při celočíselném dělení. A protože nelze provést výpočty v desetinných číslech, ale teplotu chceme zobrazit na jednu desetinu, musíme výsledek dostat desetkrát větší a toto číslo až následně zobrazit s posunutou desetinou čárkou. Upravený výpočet potom bude vypadat:

$$t = \frac{(AD0 \cdot 5 - 4096) \cdot 125}{1024} = \frac{(((((((AD0 \cdot 5 - 4096) / 4) \cdot 5) / 4) \cdot 5) / 4) \cdot 5) / 16}{10} \text{ [} \times 10^{-1} \text{ }^{\circ}\text{C]} \text{]}$$

```

$MOD841 ; definice 8052 a ADuC841
    ORG 0000h
    LJMP BEGIN ; skok na začátek programu
                ; adresace přerušení
    ORG 0003h ; interrupt IEO (vnější přerušení 0)
    ORG 000Bh ; interrupt TF0 (přerušení od časovače 0)
    ORG 0013h ; interrupt IE1 (vnější přerušení 1)
    ORG 001Bh ; interrupt TF1 (přerušení od časovače 1)
    ORG 0023h ; interrupt Serial (přerušení od sériového rozhraní)
    ORG 002Bh ; interrupt TF2 (přerušení od časovače 2)
    RETI ; konec přerušení
    ORG 0033h ; interrupt ADC (přerušení od A/D převodníku)
    MOV 0Eh,ADCDATAL ; ADC změřená hodnota – nízký byte
    MOV 0Fh,ADCDATAH ; ADC změřená hodnota – vysoký byte
    RETI
    ORG 003Bh ; interrupt SPI/I2C (přerušení od SPI nebo I2C komunikace)
    ORG 0043h ; interrupt PSM (přerušení od monitoru napájení)
    ORG 0053h ; interrupt TIC (přerušení od časového intervalového čítače)
    ORG 005Bh ; interrupt WDS (přerušení od hlídače běhu)
    RETI ; konec přerušení
    ORG 0060h
BEGIN: MOV ADCCON1,#09Ch ; zapnutí ADC, ADCLK = MCLK/4, HOLD = 8 ADCLK
    MOV ADCCON2,#0 ; nastavení kanálu A/D převodníku – vstup 0
    SETB EADC ; povolení přerušení od A/D převodníku
    SETB EA ; povolení všech přerušení
    MOV PWMCON,#021h ; nastavení režimu PWM mode 2 – frekvence = fosc/D
LOOP: MOV A,#010 ; nastavení počtu cyklů čekání 10 =>čekání 100ms
    CALL DELAY ; volání podprogramu čekání
    SETB SCONV ; start jednoho měření A/D převodníku, po skončení měření
                ; je zavoláno přerušení a výsledek je v 0Eh, 0Fh
    MOV A,#010 ; nastavení počtu cyklů čekání 10 =>čekání 100ms
    CALL DELAY ; volání podprogramu čekání
                ; výpočet z naměřené hodnoty na desetiny °C
    MOV R0,0Eh ; přesunutí obsahu čítače (nízký byte) do R0
    MOV R1,0Fh ; přesunutí obsahu čítače (vysoký byte) do R1
    CALL MUL5 ; R0,R1 krát 5 (výsledek v R0,R1, použije R2,R3,Acc)
                ; mínus 4096 (01000h) – stačí odečíst jen horní byte
    MOV A,R1 ; přesunutí obsahu R1 do akumulátoru
    CLR C ; příprava bitu přetečení (log. 0)
    SUBB A,#010h ; odečtení horního byte 4096
    MOV R1,A ; přesunutí obsahu akumulátoru do R1
    CALL DIV4 ; R0,R1 děleno 4 (výsledek v R0,R1, použije Acc)
    CALL MUL5 ; R0,R1 krát 5 (výsledek v R0,R1, použije R2,R3,Acc)
    CALL DIV4 ; R0,R1 děleno 4 (výsledek v R0,R1, použije Acc)
    CALL MUL5 ; R0,R1 krát 5 (výsledek v R0,R1, použije R2,R3,Acc)
    CALL DIV4 ; R0,R1 děleno 4 (výsledek v R0,R1, použije Acc)
    CALL MUL5 ; R0,R1 krát 5 (výsledek v R0,R1, použije R2,R3,Acc)
    CALL DIV16 ; R0,R1 děleno 16 (výsledek v R0,R1, použije Acc)
                ; v R0,R1 je teplota v desetínách °C
    JMP LOOP ; skok na začátek smyčky

```

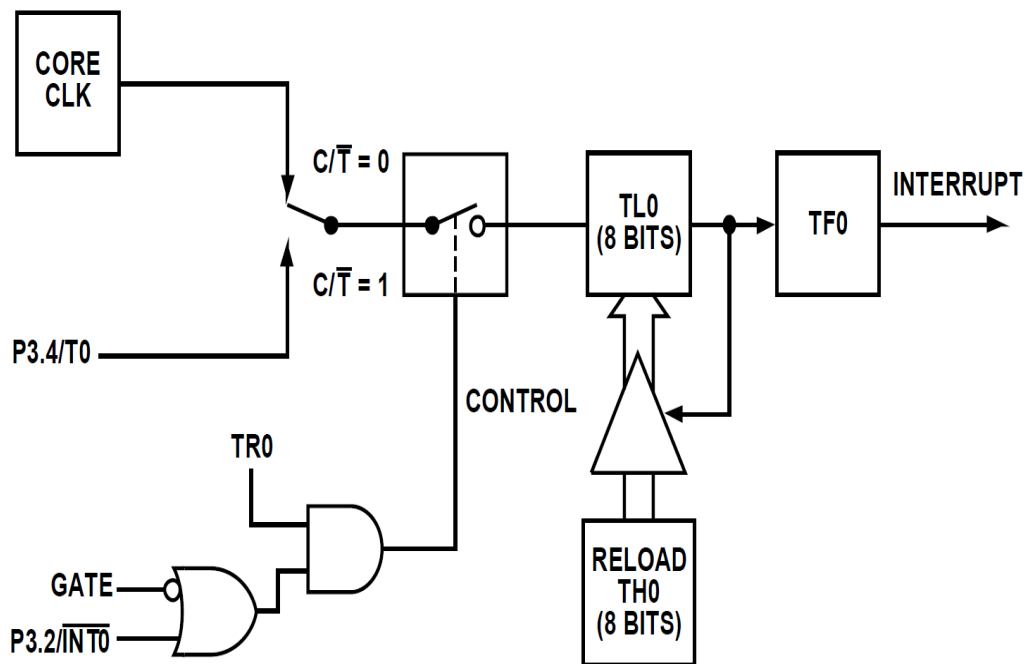
```

DELAY:                                     ; doba cyklu DLY0 na základě 11,0592MHz taktu jádra
                                           ;  $(2 + 144 \times (2 + 255 \times 3 + 3) + 3) / 11\,059\,200 = 0,01s$ 
                                           ; uložení počtu cyklů čekání z akumulátoru do R5
      MOV   R5,A                            ; nastavení zpoždění – cyklus 0, R6 = 144      (2 strojové doby)
DLY0:   MOV   R6,#090H                      ; nastavení zpoždění – cyklus 1, R7 = 255      (2 strojové doby)
DLY1:   MOV   R7,#0FFH                      ; R7=R7-1 pokud R7 není 0                       (3 strojové doby)
      DJNZ  R7,$                            ; R6=R6-1 a skok na DLY1, pokud R6 není 0      (3 strojové doby)
      DJNZ  R6,DLY1                         ; R5=R5-1 a skok na DLY0, pokud R5 není 0      (3 strojové doby)
      RET                                     ; návrat z podprogramu
DIV16:  CALL  DIV4                          ; RO,R1 děleno 4 (výsledek v RO,R1, použije Acc)
DIV4:   CALL  DIV2                          ; RO,R1 děleno 2 (výsledek v RO,R1, použije Acc)
DIV2:   MOV   A,R1                          ; přesunutí obsahu R1 do akumulátoru
      CLR   C                               ; příprava bitu přetečení (log. 0)
      RRC  A                               ; A / 2 (je použit bit přetečení)
      MOV  R1,A                             ; přesunutí obsahu akumulátoru do R1
      MOV  A,R0                             ; přesunutí obsahu R0 do akumulátoru
      RRC  A                               ; A / 2 (je použit bit přetečení)
      MOV  R0,A                             ; přesunutí obsahu akumulátoru do R0
      RET                                     ; návrat z podprogramu
MUL5:   MOV  A,R0                          ; přesunutí obsahu R0 do akumulátoru
      MOV  R2,A                             ; přesunutí obsahu akumulátoru do R2
      MOV  A,R1                             ; přesunutí obsahu R1 do akumulátoru
      MOV  R3,A                             ; přesunutí obsahu akumulátoru do R3
      CALL MUL4                            ; RO,R1 krát 4 (výsledek v RO,R1, použije Acc)
      MOV  A,R0                             ; přesunutí obsahu R0 do akumulátoru
      ADD  A,R2                             ; součet RO+R2 (Acc = 4 x vstup + vstup)
      MOV  R0,A                             ; přesunutí obsahu akumulátoru do R0
      MOV  A,R1                             ; přesunutí obsahu R1 do akumulátoru
      ADDC A,R2                             ; součet R1+R3 (Acc = 4 x vstup + vstup)
      MOV  R1,A                             ; přesunutí obsahu akumulátoru do R1
      RET                                     ; návrat z podprogramu
MUL4:   CALL  MUL2                          ; RO,R1 krát 2 (výsledek v RO,R1, použije Acc)
MUL2:   MOV  A,R0                          ; přesunutí obsahu R0 do akumulátoru
      CLR   C                               ; příprava bitu přetečení (log. 0)
      RLC  A                               ; A x 2 (je použit bit přetečení)
      MOV  R0,A                             ; přesunutí obsahu akumulátoru do R0
      MOV  A,R1                             ; přesunutí obsahu R1 do akumulátoru
      RLC  A                               ; A x 2 (je použit bit přetečení)
      MOV  R1,A                             ; přesunutí obsahu akumulátoru do R1
      RET                                     ; návrat z podprogramu
END

```

2.2.3. Interní časovač jako synchronizace běhu programu

Při tvorbě uživatelského programu je v některých případech potřeba synchronního běhu cyklu programu. To lze provést několika způsoby. Mezi nejjednodušší patří synchronizace od vnitřního časovače procesoru. Od vnitřního časovače procesoru dojde k aktivaci přerušení (interrupt). Během přerušení dojde k zastavení normálního běhu programu a začne se provádět podprogram přerušení. Je vhodné použít časovač s automatickým obnovením, kdy po dokončení časování se do časovače uloží počáteční hodnota. Toto splňuje časovač v režimu 2, proto v SFR TMOD nastavíme Gate = 0; $C/\bar{T} = 0$; M1 = 1; M0 = 0. Do TL0 a TH0 uložíme počáteční hodnotu časovače.



Obr. 2.9 Časovač / čítač v režimu 2 (8-Bit časovač / čítač s automatickým obnovením) [13]

U testovacího vývojového modulu je osazen krystal s taktovací frekvencí 11,0592 MHz, což odpovídá strojovému taktu přibližně 90,4ns. U 8-bitového časovače (čítače) lze dosáhnout maximálního času mezi přerušeními $255 \times 90,4 \text{ ns} \approx 23 \mu\text{s}$. Pro synchronizaci je tento čas velmi krátký, tak použijeme jednoduchou pomůcku, kdy v přerušení budeme čítat 8-bitové číslo. Při této pomůcce lze pak dosáhnout maximální čas $256 \times 255 \times 90,4 \text{ ns} \approx 5,9 \text{ ms}$. Pro další použití je nejvhodnější použít nějaký celočíselný čas synchronizace. Zkusím zvolit čas rovný 5 ms.

$$\frac{256 \cdot (256 - TH0)}{11059200} = 0,005 \text{ s} \rightarrow 256 \cdot (256 - TH0) = 55296 \rightarrow 256 - TH0 = 216 \rightarrow TH0 = 40$$

Při výpočtu vyšlo TH0 celé číslo, proto je zvolený čas vhodný pro přesnou synchronizaci při použití výše uvedené krystalu a nebude nutná žádná následná kompenzace. Při inicializaci časovače uložíme stejnou hodnotu i do TL0.

Částečný programový výpis s definicí interního časovače a synchronizací průběhu smyčky hlavního programu na 10 ms:

```

$MOD841                ; definice 8052 a ADuC841
    ORG    0000h
    LJMP   BEGIN        ; skok na začátek programu
                        ; adresace přerušení
    ORG    0003h        ; interrupt IE0   (vnější přerušení 0)
    ORG    000Bh        ; interrupt TF0   (přerušení od časovače 0)
    INC    0Ch          ; pomocný čítač pro takt 5ms
    SJMP   INTER_TF0    ; pozor je nutné přeskočit pevné definice přerušení
    ORG    0013h        ; interrupt IE1   (vnější přerušení 1)
    ORG    001Bh        ; interrupt TF1   (přerušení od časovače 1)
    ORG    0023h        ; interrupt Serial (přerušení od sériového rozhraní)
    ORG    002Bh        ; interrupt TF2   (přerušení od časovače 2)
    ORG    0033h        ; interrupt ADC    (přerušení od A/D převodníku)
    ORG    003Bh        ; interrupt SPI/I2C (přerušení od SPI nebo I2C komunikace)
    ORG    0043h        ; interrupt PSM    (přerušení od monitoru napájení)
    ORG    0053h        ; interrupt TIC    (přerušení od časového intervalového čítače)
    ORG    005Bh        ; interrupt WDS    (přerušení od hlídače běhu)
    RETI                    ; konec přerušení
    ORG    0060h

INTER_TF0:              ; pokračování přerušení TF0
    PUSH   Acc            ; uschování obsahu akumulátoru
    MOV    A,0Ch          ; přesunutí obsahu čítače do akumulátoru
    JNZ    $+4            ; pokud akumulátor<>0, přeskočí následující příkaz
    INC    0Dh            ; druhý pomocný časovač (+1 každých 5ms)
    POP    Acc            ; vracení původního obsahu do akumulátoru
    RETI                    ; konec přerušení

BEGIN:
    ...                    ; inicializace časovače
    MOV    TLO,#028h      ; nastavení TLO=40 (28hex)
    MOV    TH0,#028h      ; nastavení TH0=40 (28hex)
    MOV    TMOD,#022h     ; nastavení režimu časovačů
    SETB   TRO            ; start časovače 0
    SETB   EA            ; povolení všech přerušení
    ...

LOOP:                    ; začátek smyčky programu
    ...                    ; vlastní tělo vykonávaného programu
    MOV    A,#2            ; požadovaná hodnota pro synchronizaci (10ms)
SYNC:    CJNE   A,0Dh,SYNC ; čeká dokud druhý pomocný časovač = Acc
    MOV    0Dh,#0        ; 0 do pomocného časovače
    ...
    AJMP   LOOP

END

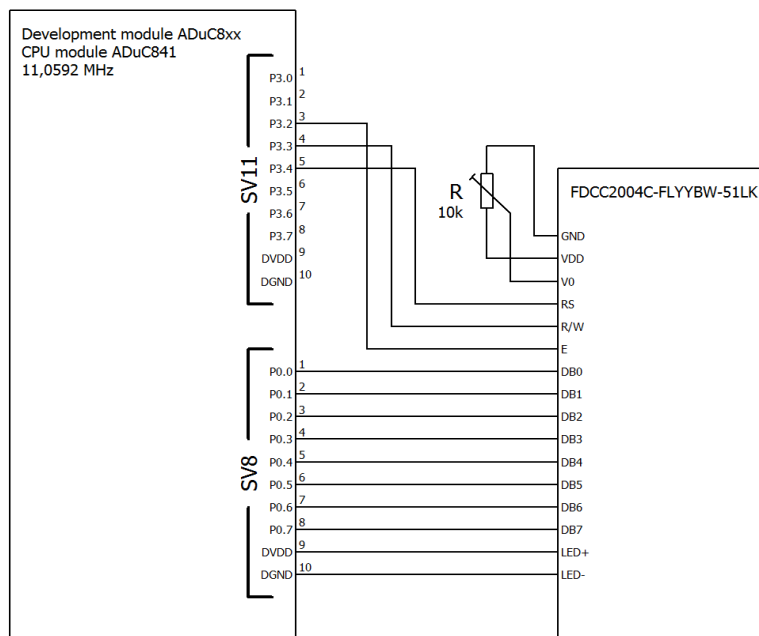
```


2.2.4. Displej 4×20 znaků (FDCC2004C – FLYYBW – 51LK)

Zobrazení stavu a nastavení hodnot, případně jiná hlášení uživateli jsou možná mimo jiné i pomocí displeje. Dále bude popsáno použití monochromatického čtyřřádkového alfanumerického maticového displeje FDCC2004C od firmy FORDATA ELECTRONIC CO.,LTD.[17] Řadič tohoto displeje je kompatibilní s běžně užívaným řadičem HD44780, který je ovládaný třemi řídicími vstupy a čtyřmi nebo osmi datovými signály (ovládání řadiče viz Tab. 2.1 – čerpáno z [17]):

Význam	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Délka
smaže disp. a nastaví kurzor na začátek	0	0	0	0	0	0	0	0	0	1	1,52ms
nastaví kurzor na začátek	0	0	0	0	0	0	0	0	1	x	1,52ms
směr posuvu kurzoru I/D (0=vlevo, 1=vpravo), posuv textu S (0=ne, 1=ano)	0	0	0	0	0	0	0	1	I/D	S	38μs
D - zapne displej, C - zapne kurzor, B - zapne blikání kurzoru	0	0	0	0	0	0	1	D	C	B	38μs
1× posune (S/C=0 kurzor, S/C=1 text) směrem (R/L=0 vlevo, R/L=1 vpravo)	0	0	0	0	0	1	S/C	R/L	x	x	38μs
inicializace: DL=0 4-bit, DL=1 8-bit mód, N=0 jednořádkový, N=1 dvouřádkový disp., F=0 font 5×8 bodů, F=1 font 5×10 bodů	0	0	0	0	1	DL	N	F	x	x	38μs
přepnutí na zápis do CGRAM	0	0	0	1	adresa v CGRAM					38μs	
přepnutí na zápis do DDRAM	0	0	1	adresa v DDRAM					38μs		
čtení příznaku BF (BF=0 příjem povolen, BF=1 řadič zaneprázdněn), čtení adresy v DDRAM	0	1	BF	adresa v DDRAM					0μs		
zápis dat do CGRAM nebo DDRAM	1	0	data					38μs			
čtení dat z CGRAM nebo DDRAM	1	1	data					38μs			

Tab. 2.1: Řídicí instrukce displeje s řadičem HD44780 [17]



Obr. 2.10: Propojení vývojového modulu a displeje

Na displeji jsou osazeny propojky J2, J3, J5 a J6. Těmito propojkami je spojeno napájení pro vlastní displej a podsvětlení displeje, proto není potřeba napájet displej zvlášť. Vývod V0 (nastavení kontrastu) připojíme pomocí odporového děliče, vytvořeného z trimru 10 až 20 kΩ vloženého mezi VDD a GND. Takto propojený displej je možné ovládat v 8-bitovém módu.

Čtyřřádkový monochromatický alfanumerický maticový displej FDCC2004C s řadičem se chová jako dvouřádkový displej se čtyřiceti sloupci, kde první dvě řádky s dvaceti znaky pokračují na dalších dvou řádkách dalšími dvaceti znaky.

Definice vstup-výstupů ovládání displeje:

DB0	BIT	P0.0	; data bit 0
DB1	BIT	P0.1	; data bit 1
DB2	BIT	P0.2	; data bit 2
DB3	BIT	P0.3	; data bit 3
DB4	BIT	P0.4	; data bit 4
DB5	BIT	P0.5	; data bit 5
DB6	BIT	P0.6	; data bit 6
DB7	BIT	P0.7	; data bit 7
DADI	DATA	P0	; data 8-bitově
E	BIT	P3.2	; hodinový takt
RW	BIT	P3.3	; volba zápis (log.0) / čtení (log.1)
RS	BIT	P3.4	; volba instrukce (log.0) / data (log.1)

Makroinstrukce ADDR – pro nastavení kurzoru displeje na řádek, kde řádky 0 až 3 se zadávají do parametru ROW a sloupec do parametru COL.

```

ADDR  MACRO ROW, COL      ; nastavit kurzor na řádek ROW a sloupec COL
      IF ROW=0            ; pokud ROW=0 provede se následující
      MOV  A,#80h+COL     ; hodnota adresy začátku řádku 0 do akumulátoru
      ENDIF
      IF ROW=1            ; pokud ROW=1 provede se následující
      MOV  A,#0C0h+COL    ; hodnota adresy začátku řádku 1 do akumulátoru
      ENDIF
      IF ROW=2            ; pokud ROW=2 provede se následující
      MOV  A,#94h+COL     ; hodnota adresy začátku řádku 2 do akumulátoru
      ENDIF
      IF ROW=3            ; pokud ROW=3 provede se následující
      MOV  A,#0D4h+COL    ; hodnota adresy začátku řádku 3 do akumulátoru
      ENDIF
      ACALL CLOCKR        ; podprogram – nastavit kurzor na adresu v Acc
      ENDM

```

Pomocná makroinstrukce ADDR2 – pro nastavení kurzoru displeje na začátek řádku. Pokud je zadána hodnota parametru ROW v rozsahu 0 až 3 provede nastavení na začátek konkrétního řádku displeje. Při zadané hodnotě 4 bude nastaven kurzor na začátek řádku 0 a při hodnotě 5 bude nastaven na začátek řádku 1. Do parametru R0 je uložena příslušná hodnota počtu sloupců pro makroinstrukce CLRR nebo TXTR.

```

ADDR2  MACRO ROW          ; nastavit kurzor na začátek řádku ROW
      ; (0 – 3 jeden řádek ROW; 4 dva řádky 0 a 2; 5 dva řádky 1 a 3)
      IF ROW<4            ; pokud ROW<4 provede se následující
      ADDR  ROW, 0        ; nastavit ukazovátka na začátek řádky ROW
      MOV   R0,#20        ; počet sloupců pro CLRR nebo TXTR (20 sloupců)
      ENDIF
      IF ROW>3            ; pokud R>3 provede se následující
      ADDR  ROW-4, 0      ; nastavit ukazovátka na začátek řádky (ROW-4)
      MOV   R0,#40        ; počet sloupců pro CLRR nebo TXTR (2řádky × 20 sloupců)
      ENDIF
      ENDM

```

Makroinstrukce CLRR – smazání řádky displeje, význam hodnoty ROW viz makroinstrukce ADDR2.

CLRR	MACRO	ROW	; smazat řádek ROW
			; (0 – 3 smaže jeden řádek ROW; 4 smaže dva řádky 0 a 2; 5 smaže dva řádky 1 a 3)
	ADDR2	ROW	; nastavit kurzor na začátek řádky ROW
	ACALL	CLEARR	; podprogram – pošle R0 krát znak mezery (použije a poškodí R0)
	ENDM		

Makroinstrukce TXTR – vypsání textu na displej, význam hodnoty ROW viz makroinstrukce ADDR2.

Za makroinstrukcí TXTR je nutné vložit text který se bude vypisovat. Délka textu musí odpovídat hodnotě v registru R0 z makroinstrukce ADDR2.

TXTR	MACRO	ROW	; vypisovat text na řádek R (text za makroinstrukcí)
			; (0 – 3 text na jeden řádek ROW; 4 text na dva řádky 0 a 2; 5 text na dva řádky 1 a 3)
	ADDR2	ROW	; nastavit kurzor na začátek řádky ROW
	MOV	R1,#1	; ukazovátka na text za makroinstrukcí (R1)
	MOV	A,R1	; hodnota ukazovátka (R1) do akumulátoru
	ACALL	\$+8	; podprogram – načítá znaky textu z pozice ukazovátka do Acc
	ACALL	CLOCKT	; podprogram – Acc odešle do řadiče displeje; R1 = R1 + 1; takt E
	DJNZ	R0,\$-5	; opakuje se dokud se nenačtou všechny znaky textu
	IF	ROW<4	; pokud ROW<4 provede se následující
	SJMP	\$+24	; skok za poslední znak textu
	ENDIF		
	IF	ROW>3	; pokud ROW>3 provede se následující
	SJMP	\$+44	; skok za poslední znak textu
	ENDIF		
	MOVC	A,@A+PC	; načítá znaky textu z pozice ukazovátka do Acc ((A + PC) → A)
	RET		; návrat z podprogramu
	ENDM		

Inicializace a počáteční nastavení displeje, inicializace a nastavení časovačů může být shodné jako v programu interní synchronizace popsány v kapitole 2.2.3:

INIC:	...		; inicializace časovačů, přerušení ...
	MOV	DADI,#0	; data pro displej (0)-
	CLR	RS	; volba významu datových vodičů; log 0 – řídicí slovo; log 1 – data
	CLR	RW	; volba směru přenosu; log 0 – zápis; log 1 – čtení
	CLR	E	; hodinový takt – náběžná hrana potvrzuje přenos
	MOV	A,#080h	; požadovaná hodnota pro synchronizaci (2,5ms)
INIC1:	CJNE	A,0Ch,INIC1	; čeká 2,5ms po startu
	MOV	DADI,#038h	; řídicí slovo (8-bitový mód, dvouřádkový displej, font 5x8 bodů)
	ACALL	CLOCK	; podprogram – provede jeden hodinový takt na výstupu E
	MOV	DADI,#0Ch	; řídicí slovo (zap. displeje, vypnutí zobrazení a blikání kurzoru)
	ACALL	CLOCK	; podprogram – provede jeden hodinový takt na výstupu E
	MOV	DADI,#0	; konec inicializace displeje
	ACALL	CLOCK	; podprogram – provede jeden hodinový takt na výstupu E
	TXTR	4	; makroinstrukce – vypíše řádek 0 a 2
	DB	'***** IOP-04 v. 2.00 **'	
	TXTR	5	; makroinstrukce – vypíše řádek 1 a 3
	DB	'*** J. Vávra (c) *****'	
	MOV	A,#200	; požadovaná hodnota pro synchronizaci (200 x 5ms = 1s)
INIC2:	CJNE	A,0Dh,INIC2	; čeká dokud druhý pomocný časovač = Acc (inicializační nápis)
	CLRR	4	; makroinstrukce – smaže řádek 0 a 2
	CLRR	5	; makroinstrukce – smaže řádek 1 a 3
	...		; další pokračování programu

Podprogramy využívané makroinstrukcemi a při inicializaci:

```

CLEARR:
    MOV    DADI,#20h      ; data pro displej (mezera)
    ACALL  CLOCK          ; podprogram – provede jeden hodinový takt na výstupu E
    DJNZ  RO,CLEARR      ; R0=R0-1 a skok na CLEARR, pokud R0 není 0
    RET                    ; návrat z podprogramu
CLOCKR: CLR    RS        ; volba významu datových vodičů; log 0 – řídicí slovo
CLOCKT: MOV    DADI,A    ; řídicí slovo z akumulátoru
        INC    R1        ; R1=R1+1 – využívá TXTR jako ukazovátka v textu
CLOCK:  SETB   E         ; výstup E (hodinový takt) do log. 1
        PUSH  B         ; uschování obsahu registru B
        MOV   B,#84     ; požadovaná hodnota pro mezeru (84 × 3 strojové doby)
        DJNZ B,$        ; mezera 252 strojových dob
        CLR   E         ; výstup E (hodinový takt) do log. 0
        MOV   B,#72     ; požadovaná hodnota pro mezeru (72 × 3 strojové doby)
        DJNZ B,$        ; mezera 216 strojových dob
        POP   B         ; vrácení původního obsahu do registru B
        SETB  RS        ; volba významu datových vodičů; log 1 – data
        RET                    ; návrat z podprogramu

```

Text je možné vypisovat na displej pomocí makroinstrukce TXTR jak je vidět v části programu inicializace. Pro úplnost a lepší pochopení adresace řádek následují dvě ukázky včetně grafické podoby displeje:

```

; Ukázka programu 1:
    TXTR  4                ; makroinstrukce – vypíše řádek 0 a 2
    DB    '***** IOP-04 v. 2.00 **'
    TXTR  5                ; makroinstrukce – vypíše řádek 1 a 3
    DB    '*** J. Vavra (c) *****'
; jiný zápis programu (programově i strojově delší):
    TXTR  0                ; makroinstrukce – vypíše řádek 0
    DB    '*****'
    TXTR  1                ; makroinstrukce – vypíše řádek 1
    DB    '*** J. Vavra (c) ***'
    TXTR  2                ; makroinstrukce – vypíše řádek 2
    DB    '** IOP-04 v. 2.00 **'
    TXTR  3                ; makroinstrukce – vypíše řádek 3
    DB    '*****'
; Na displeji bude zobrazeno:
; řádek 0:      *****
; řádek 1:      *** J. Vavra (c) ***
; řádek 2:      ** IOP-04 v. 2.00 **
; řádek 3:      *****
; Ukázka programu 2:
    TXTR  4                ; makroinstrukce – vypíše řádek 0 a 2
    DB    '0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcd'
    TXTR  5                ; makroinstrukce – vypíše řádek 1 a 3
    DB    'efghijklmnopqrstuvwxyz{[*+==/?.,!@#%&]} '
; Na displeji bude zobrazeno:
; řádek 0:      0123456789ABCDEFGHIJ
; řádek 1:      efghijklmnopqrstuvwxyz
; řádek 2:      KLMNOPQRSTUVWXYZabcd
; řádek 3:      {[*+==/?.,!@#%&]}

```

Číselnou hodnotu je možné zobrazit pomocí následujícího podprogramu ZOIN, před zavoláním podprogramu je nutné nastavit pozici vypisování čísla pomocí makroinstrukce ADDR. Vlastní zobrazované číslo je potřeba uložit do registrů R2, R3 v dekadickém formátu. Každé čtyři bity odpovídají jedné číslici. Číslice se budou zobrazovat v pořadí: horní polovina R3, dolní polovina R3, horní polovina R2 a nakonec dolní polovina R2. Hodnota pomocného registru R1 určuje pozici desetinné čárky. Význam hodnot R1 je patrný z výpisu programu.

```

;R2 <= low cisla   (12h)           R3 <= high cisla   (34h)
;R1 <= pozice desetinné čárky (hodnota se průběhem změní !)
;R1 = 0           X,XXX           ( 1,234 )
;R1 = 1           XX,XX           ( 12,34 )
;R1 = 2           XXX,X           ( 123,4 )
;R1 = 3           XXXX           ( 1234 )
ZOIN:  MOV     A,R1                ; obsah registru R1 do akumulátoru
       JNZ     ZO0                 ; skok pokud není akumulátor nulový
       MOV     A,#0F0h             ; maska přenosu do akumulátoru
       ANL     A,R3                ; maskovaný přenos obsahu R3 do akumulátoru
       JNZ     ZO1                 ; skok pokud není akumulátor nulový
       MOV     A,#' '              ; znak mezery do akumulátoru
       ACALL   CLOCKT              ; podprogram – Acc odešle do řadiče displeje; R1 = R1 + 1; takt E
       MOV     A,#0Fh              ; maska přenosu do akumulátoru
       ANL     A,R3                ; maskovaný přenos obsahu R3 do akumulátoru
       JNZ     ZO2                 ; skok pokud není akumulátor nulový
       MOV     A,#' '              ; znak mezery do akumulátoru
       ACALL   CLOCKT              ; podprogram – Acc odešle do řadiče displeje; R1 = R1 + 1; takt E
       MOV     A,#0F0h             ; maska přenosu do akumulátoru
       ANL     A,R2                ; maskovaný přenos obsahu R2 do akumulátoru
       JNZ     ZO3                 ; skok pokud není akumulátor nulový
       MOV     A,#' '              ; znak mezery do akumulátoru
       ACALL   CLOCKT              ; podprogram – Acc odešle do řadiče displeje; R1 = R1 + 1; takt E
       AJMP    ZO4                 ; skok na zobrazení poslední číslice
ZO0:   MOV     A,#0F0h             ; maska přenosu do akumulátoru
       ANL     A,R3                ; maskovaný přenos obsahu R3 do akumulátoru
ZO1:   SWAP    A                   ; výměna horních a dolních 4 bitů v akumulátoru mezi sebou
       ORL     A,#30h              ; převod na ASCII číslo (přičtení 30h)
       ACALL   CLOCKT              ; podprogram – Acc odešle do řadiče displeje; R1 = R1 + 1; takt E
       MOV     A,R1                ; obsah registru R1 do akumulátoru
       DEC     A                   ; kontrola na desetinou čárku
       JNZ     $+6                 ; skok pokud je akumulátor nulový
       MOV     A,#','              ; znak čárky do akumulátoru
       ACALL   CLOCKT              ; podprogram – Acc odešle do řadiče displeje; R1 = R1 + 1; takt E
       MOV     A,#0Fh              ; maska přenosu do akumulátoru
       ANL     A,R3                ; maskovaný přenos obsahu R3 do akumulátoru
ZO2:   ORL     A,#30h              ; převod na ASCII číslo (přičtení 30h)
       ACALL   CLOCKT              ; podprogram – Acc odešle do řadiče displeje; R1 = R1 + 1; takt E
       MOV     A,R1                ; obsah registru R1 do akumulátoru
       DEC     A                   ; kontrola na desetinou čárku
       JNZ     $+6                 ; skok pokud je akumulátor nulový
       MOV     A,#','              ; znak čárky do akumulátoru
       ACALL   CLOCKT              ; podprogram – Acc odešle do řadiče displeje; R1 = R1 + 1; takt E
       MOV     A,#0F0h             ; maska přenosu do akumulátoru
       ANL     A,R2                ; maskovaný přenos obsahu R2 do akumulátoru

```

Z03:	SWAP	A	; výměna horních a dolních 4 bitů v akumulátoru mezi sebou
	ORL	A,#30h	; převod na ASCII číslo (přičtení 30h)
	ACALL	CLOCKT	; podprogram – Acc odešle do řadiče displeje; R1 = R1 + 1; takt E
Z04:	MOV	A,R1	; obsah registru R1 do akumulátoru
	DEC	A	; kontrola na desetinou čárku
	JNZ	\$+6	; skok pokud je akumulátor nulový
	MOV	A,#','	; znak čárky do akumulátoru
	ACALL	CLOCKT	; podprogram – Acc odešle do řadiče displeje; R1 = R1 + 1; takt E
	MOV	A,#0Fh	; maska přenosu do akumulátoru
	ANL	A,R2	; maskovaný přenos obsahu R2 do akumulátoru
	ORL	A,#30h	; převod na ASCII číslo (přičtení 30h)
	ACALL	CLOCKT	; podprogram – Acc odešle do řadiče displeje; R1 = R1 + 1; takt E
	RET		

2.2.5. Pulsně šířková modulace (PWM)

Pulsně šířkovou modulaci je možné použít na řízení výkonu nebo otáček motoru, jasu zdroje světla a mnoho dalších aplikací. V mikrokontroléru ADuC841 jsou implementovány dva PWM výstupy. Podrobnosti k principu a nastavení jsou v kapitole 1.2.4. Pro příklad jsem vytvořil program s použitím PWM v režimu 2 (Obr. 1.4). Pokud budou nastavené stejné hodnoty $PWM1L = PWM0H$ a $PWM0L = PWM1H$, budou výstupy P2.6 a P2.7 v protifázi. Od začátku PWM cyklu do dosažení hodnoty PWM0L bude výstup P2.6 v log. 1 a výstup P2.7 v log. 0. Po zbytek cyklu budou stavy výstupů opačné.

```

$MOD841                ; definice 8052 a ADuC841
    ORG    0000h
    LJMP   BEGIN        ; skok na začátek programu
    ORG    000Bh        ; interrupt TF0 (přerušeni od časovače 0)
    INC    0Ch          ; pomocný čítač pro takt 5ms
    SJMP   INTER_TF0    ; je nutné přeskočit pevné definice přerušeni
    ORG    0060h

INTER_TF0:              ; pokračování přerušeni TF0
    PUSH  Acc           ; uschování obsahu akumulátoru
    MOV   A,0Ch         ; přesunutí obsahu čítače do akumulátoru
    JNZ   $+4           ; pokud akumulátor<>0 přeskočí následující příkaz
    INC   0Dh           ; druhý pomocný časovač (+1 každých 5ms)
    POP   Acc           ; vracení původního obsahu do akumulátoru
    RETI                ; konec přerušeni

BEGIN:  MOV   TLO,#028h ; nastavení TLO=40 (28hex) – 5ms/256
        MOV   TH0,#028h ; nastavení TH0=40 (28hex)
        MOV   TMOD,#022h ; nastavení režimu časovačů
        SETB  TR0       ; start časovače 0
        SETB  EA        ; povolení všech přerušeni
        MOV   CFG841,#010h ; nastavení konfiguračního registru (D=512)
        MOV   PWM0L,#00h ; prvotní nastavení PWM
        MOV   PWM1H, PWM0L ; PWM1H = PWM0L
        MOV   PWM1L,#0C8h ; prvotní nastavení PWM (200 dílků) – 200 × 10ms = 2s
        MOV   PWM0H, PWM1L ; PWM0H = PWM1L
        MOV   PWMCON,#021h ; nastavení režimu PWM mode 2 – frekvence = fosc / D


LOOP1:  MOV   A,#2       ; požadovaná hodnota pro synchronizaci (10ms)
SYNC1:  CJNE  A,0Dh,SYNC1 ; čeká dokud druhý pomocný časovač = Acc
        MOV   0Dh,#0     ; 0 do pomocného časovače
        INC   PWM0L       ; prodlužuje se sepnutí P2.6
        MOV   PWM1H, PWM0L ; PWM1H = PWM0L
        MOV   A, PWM1L    ; maximální hodnota, kam až se může číst PWM0L
        CJNE  A,PWM0L,LOOP1 ; skočí na začátek smyčky 1, pokud není trvale sepnut P2.6

LOOP2:  MOV   A,#2       ; požadovaná hodnota pro synchronizaci (10ms)
SYNC2:  CJNE  A,0Dh,SYNC2 ; čeká dokud druhý pomocný časovač = Acc
        MOV   0Dh,#0     ; 0 do pomocného časovače
        DEC   PWM0L       ; zkracuje se sepnutí P2.6
        MOV   PWM1H, PWM0L ; PWM1H = PWM0L
        MOV   A,#0       ; minimální hodnota, kam až se může číst PWM0L
        CJNE  A,PWM0L,LOOP2 ; skočí na začátek smyčky 2, pokud není trvale vypnut P2.6
        AJMP  LOOP1      ; skočí na začátek smyčky 1

END

```

Závěr

Diplomová práce se věnuje problematice mikrokontrolérů řady ADuC8xx firmy . V první části práce je teoretický přehled funkcí mikrokontrolérů odchylojících se od základní řady INTEL 8052. Tato část je parafráze a volný překlad katalogových listů [5], [6], [7], [8], [9], [10], [11], [12], [13], [14] a také jsem zde čerpal ze svých dlouholetých osobních zkušeností s mikrokontroléry ATMEL 8051.

V druhé části práce jsem vytvořil několik ukázkových programů a provedl testy funkčních celků na vývojovém kitu vytvořeném jako praktická část mojí bakalářské práce, osazený mikrokontrolérem ADuC841 s krystalem s taktovací frekvencí 11,0592 MHz. V rámci řešení vlastní části diplomové práce jsem změřil proud odebíraný mikrokontrolérem ADuC841. V normálním režimu mikrokontroléru jsem naměřil proud 26,5 mA a v režimu se sníženou spotřebou jsem naměřil proud 6,1 mA. Celkový čas cyklu průběhu testovaného programu byl 25,7 s a z toho 5 s byl mikrokontrolér v normálním režimu. Průměrný proud odebíraný mikrokontrolérem byl 10,1 mA. Pokud se bude tato aplikace napájet akumulátorem s kapacitou 1 Ah při použití režimu se sníženou spotřebou bude možné provozovat mikrokontrolér 4 dny (tj. 99 hod. 19 min.). Naopak bez aktivace režimu se sníženou spotřebou je možné provozovat mikrokontrolér jen 1,5 dne (tj. 37 hod. 44 min.).

Také jsem vytvořil ukázkový program obsluhy teplotního senzoru TMP36, který je připojený na interní A/D převodník. Další vytvořenou aplikací je připojení a programová obsluha monochromatického čtyřřádkového alfanumerického maticového displeje, který umožňuje zobrazení stavu a nastavení hodnot, případně jiná (např. poruchová) hlášení uživateli.

U všech programů uvedených v této diplomové práci lze očekávat, že budou kompatibilní i s ostatními mikrokontroléry řady ADuC8xx, v případě že obsahují uvedené vlastnosti a funkční bloky mají osazené (viz přílohy B a C). Při využití krystalu s jinou frekvencí je potřeba upravit časové konstanty.

Použitá literatura

- [1] Klůčik, J.; Fronc, V.: Mikrokontroléry ATMEL s jádrem 8051, BEN – technická literatura, Praha 2001
- [2] Matoušek, D.: Práce s Mikrokontroléry Atmel AT89C2051, BEN - technická literatura, Praha 2002
- [3] Matoušek, D.: Práce s Mikrokontroléry Atmel AT89S8252, BEN - technická literatura, Praha 2002
- [4] 8051 Microcontroller Instruction Set [Online] [Download 27. 03. 2012]
<http://www.atmel.com/Images/doc0509.pdf>
- [5] ADUC812 Data Sheet Rev E, 04/2003 [Online] [Download 24. 11. 2011]
http://www.analog.com/static/imported-files/data_sheets/ADUC812.pdf
- [6] ADUC814 Data Sheet Rev A, 12/2003 [Online] [Download 21. 04. 2012]
http://www.analog.com/static/imported-files/data_sheets/ADUC814.pdf
- [7] ADUC816 Data Sheet Rev A, 01/2010 [Online] [Download 23. 04. 2012]
http://www.analog.com/static/imported-files/data_sheets/ADUC816.pdf
- [8] ADUC824 Data Sheet Rev B, 05/2002 [Online] [Download 21. 04. 2012]
http://www.analog.com/static/imported-files/data_sheets/ADUC824.pdf
- [9] ADUC831 Data Sheet Rev 0, 11/2002 [Online] [Download 10. 11. 2011]
http://www.analog.com/static/imported-files/data_sheets/ADUC831.pdf
- [10] ADUC832 Data Sheet Rev 0, 11/2002 [Online] [Download 27. 04. 2012]
http://www.analog.com/static/imported-files/data_sheets/ADUC832.pdf
- [11] ADUC834 Data Sheet Rev A, 04/2003 [Online] [Download 21. 04. 2012]
http://www.analog.com/static/imported-files/data_sheets/ADUC834.pdf
- [12] ADUC836 Data Sheet Rev A, 04/2003 [Online] [Download 21. 04. 2012]
http://www.analog.com/static/imported-files/data_sheets/ADUC836.pdf
- [13] ADUC841/ ADUC842/ADUC843 Data Sheet Rev 0, 11/2003 [Online] [Download 04. 04. 2012]
http://www.analog.com/static/imported-files/data_sheets/ADUC841_842_843.pdf
- [14] ADUC845/ ADUC847/ADUC848 Data Sheet Rev B, 02/2005 [Online] [Download 21. 04. 2012]
http://www.analog.com/static/imported-files/data_sheets/ADUC845_847_848.pdf
- [15] Analog Microcontrollers | Processors and DSP | Analog Devices [Online] [Citace 23. 04. 2012]
<http://www.analog.com/en/processors-dsp/analog-microcontrollers/processors/index.html>
- [16] Low Voltage Temperature Sensors TMP35/TMP36/TMP37 [Online] [Download 19. 04. 2013]
http://www.analog.com/static/imported-files/data_sheets/TMP35_36_37.pdf
- [17] SPECIFICATION CHARACTER TYPE DOT MATRIX LCD MODULE FDCC2004C-FLYYBW-51LK [Online] [Download 16. 04. 2012] <http://www.farnell.com/datasheets/653686.pdf>
- [18] STK500 – Atmel Corporation [Online] [Citace 07. 05. 2012] <http://www.atmel.com/tools/STK500.aspx>
- [19] Ukázkové programy pro ADuC841 [Online] [Download 22. 03. 2012]
ftp://ftp.analog.com/pub/www/technology/dataConverters/microconverter/841code_asm.zip
- [20] Windows Serial Downloader [Online] [Download 24. 09. 2011]
ftp://ftp.analog.com/pub/MicroConverter/ADuC8xx/WSD/v7.0/wsd_setup_v705.exe

Seznam obrázků

Obr. 1.1 Architektura paměti programu (8 kB interní paměť) [5], [6], [7], [8], [9], [10], [11], [12], [13], [14].....	12
Obr. 1.2 Architektura paměti dat [5], [6], [7], [8], [9], [10], [11], [12], [13], [14].....	12
Obr. 1.3 PWM režim 1 [9], [10], [11], [12], [13], [14].....	14
Obr. 1.4 PWM režim 2 [9], [10], [11], [12], [13], [14].....	14
Obr. 1.5 PWM režim 3 [9], [10], [11], [12], [13], [14].....	14
Obr. 1.6 PWM režim 4 [9], [10], [11], [12], [13], [14].....	15
Obr. 1.7 PWM režim 5 [9], [10], [11], [12], [13], [14].....	16
Obr. 1.8 PWM režim 6 [9], [10], [11], [12], [13], [14].....	16
Obr. 1.9 SPI časování, všechny druhy provozu [5], [6], [7], [8], [9], [10], [11], [12], [13], [14].....	17
Obr. 1.10 Bloková schéma časového intervalového čítače [5], [6], [7], [8], [9], [10], [11], [12], [13], [14].....	19
Obr. 2.1: Rozmístění součástek a popis přední strany vývojového kitu ADuC8xx.....	20
Obr. 2.2: Schema zapojení vývojového kitu ADuC8xx.....	21
Obr. 2.3 Doporučené propojení sériové komunikace a PSEN.....	22
Obr. 2.4 Okno aplikace Windows Serial Downloader.....	22
Obr. 2.5 Schéma zapojení měření spotřeby mikrokontroléru.....	25
Obr. 2.6: Graf průběhu proudu mikrokontroléru v normálním režimu včetně spínání LED.....	26
Obr. 2.7: Graf průběhu proudu celého cyklu mikrokontroléru bez/s LED.....	26
Obr. 2.8: Závislost výstupního napětí snímače na teplotě [16].....	28
Obr. 2.9 Časovač / čítač v režimu 2 (8-Bit časovač / čítač s automatickým obnovením) [13].....	31
Obr. 2.10: Propojení vývojového modulu a displeje.....	33
Obr. A.1 Blokové schéma ADuC812, ADuC831, ADuC832, ADuC841, ADuC842 a ADuC843 (šedě vybarvené bloky nejsou součástí ADuC812), blok DAC není osazen v ADuC843 a blok PLL jen v ADuC832, ADuC842 a ADuC843 [5], [9], [10], [13].....	45
Obr. A.2: Blokové schéma ADuC814 [6].....	45
Obr. A.3: Blokové schéma ADuC816, ADuC836 (šedě vybarvené bloky nejsou součástí ADuC816) [7], [12]....	46
Obr. A.4: Blokové schéma ADuC824, ADuC834 (šedě vybarvené bloky nejsou součástí ADuC824) [8], [11]....	46
Obr. A.5: Blokové schéma ADuC845, ADuC847 a ADuC848, Auxiliary (pomocný) A/D převodník jen v ADuC845, v ADuC848 16bitový převodník [14].....	47
Obr. B.1: Klíč k SFR tabulkám [5], [6], [7], [8], [9], [10], [11], [12], [13], [14].....	48
Obr. B.2: Tabulka SFR ADuC812 [5].....	48
Obr. B.3: Tabulka SFR ADuC814 [6].....	49
Obr. B.4: Tabulka SFR ADuC816 [7].....	49
Obr. B.5: Tabulka SFR ADuC824 [8].....	50
Obr. B.6: Tabulka SFR ADuC831 [9].....	50
Obr. B.7: Tabulka SFR ADuC832 [10].....	51
Obr. B.8: Tabulka SFR ADuC834 [11].....	51
Obr. B.9: Tabulka SFR ADuC836 [12].....	52
Obr. B.10: Tabulka SFR ADuC841, ADuC842 a ADuC843 [13].....	52
Obr. B.11: Tabulka SFR ADuC845, ADuC847 a ADuC848 [14].....	53

Seznam tabulek

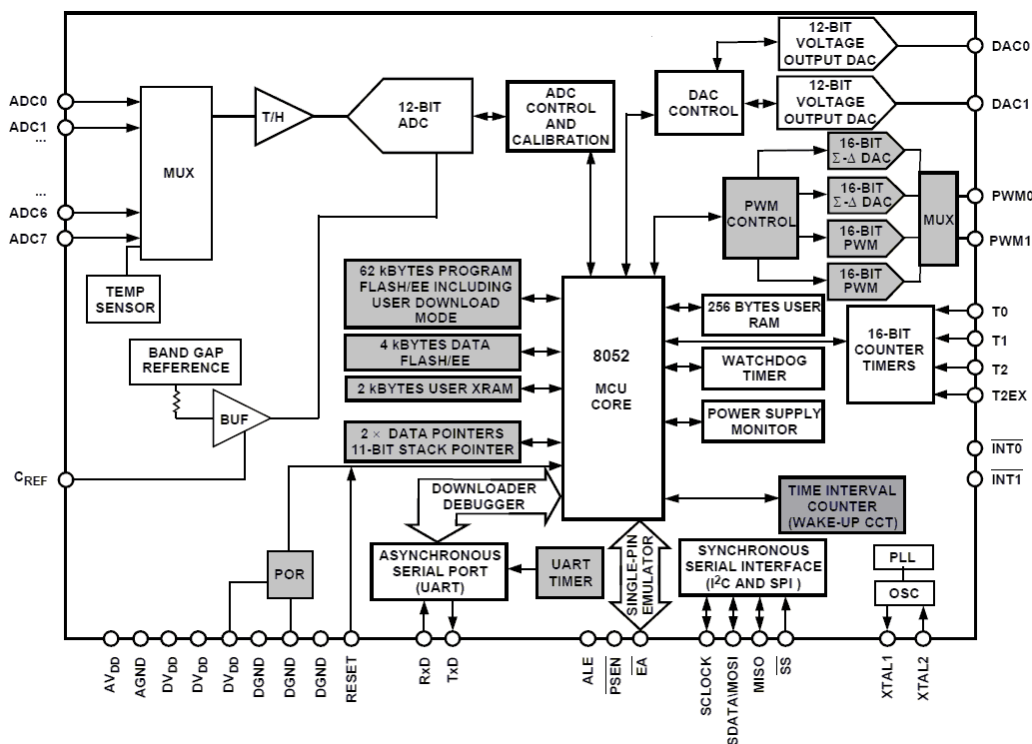
Tab. 1.1 Seznam mikrokontrolérů ADuC8xx a jejich základní parametry [15].....	11
Tab. 1.2 Vektorové adresy přerušení [5], [6], [7], [8], [9], [10], [11], [12], [13], [14].....	11
Tab. 2.1: Řídící instrukce displeje s řadičem HD44780 [17].....	33

Seznam příloh

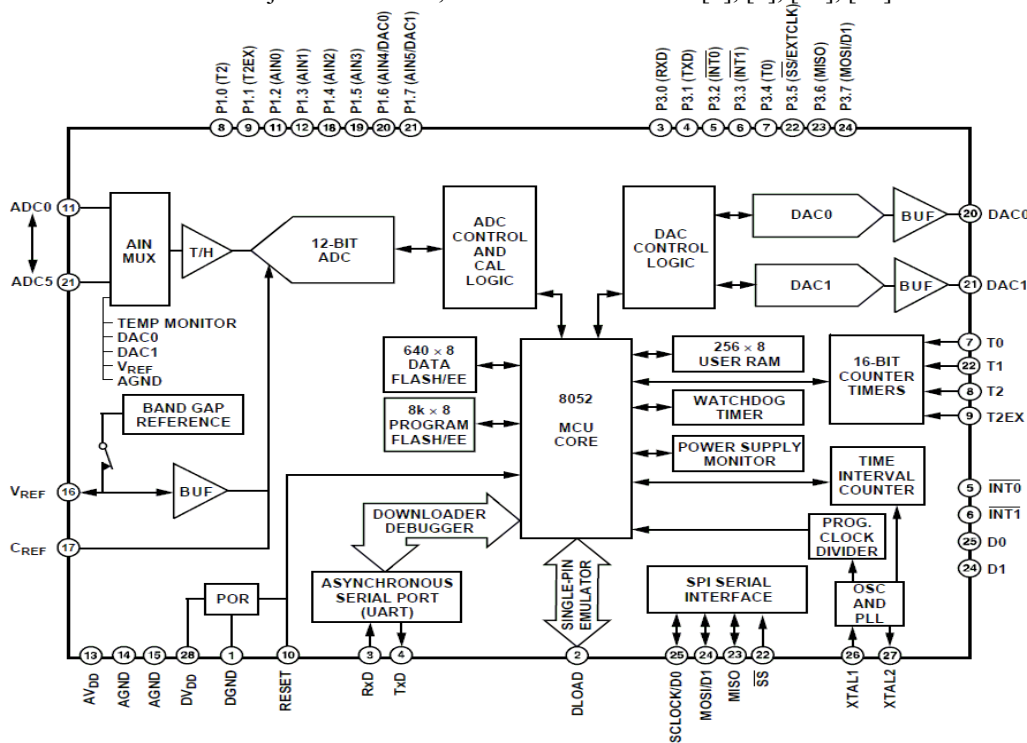
PŘÍLOHA A –BLOKOVÉ SCHÉMA ADUC8XX.....	45
PŘÍLOHA B –SPECIAL FUNCTION REGISTERS (SFR) OBLAST SPECIÁLNÍCH REGISTRŮ.....	48
PŘÍLOHA C –PARAMETRY SFR ADUC8XX.....	54
C.1.ADC0CON (PRIMARY ADC CONTROL REG. – ŘÍDÍCÍ SLOVO PRIMÁRNÍHO A/D).....	54
C.2.ADC0H/M/(L) (PRIMARY ADC CONVERSION RESULT REG. – DATA Z PRIM. A/D).....	54
C.3.ADC1CON (AUXILIARY ADC CONTROL REG. – ŘÍDÍCÍ SLOVO POMOCNÉHO A/D).....	54
C.4.ADC1H/L (AUXILIARY ADC CONVERSION RESULT REG. – DATA Z POMOCN. A/D).....	55
C.5.ADCCON1 (ADC CONTROL – ŘÍDÍCÍ SLOVO 1 A/D PŘEVODNÍKU).....	55
C.6.ADCCON2 (ADC CONTROL – ŘÍDÍCÍ SLOVO 2 A/D PŘEVODNÍKU).....	56
C.7.ADCCON3 (ADC CONTROL – ŘÍDÍCÍ SLOVO 3 A/D PŘEVODNÍKU).....	58
C.8.ADCDATAH/L (ADC DATA REGISTERS – DATA Z A/D PŘEVODNÍKU).....	58
C.9.ADCGAINH/L (ADC GAIN CALIBRATION COEFFICIENT – ZISK A/D PŘEVODNÍKU).....	58
C.10.ADCMODE (ADC MODE REGISTER – REŽIM A/D PŘEVODNÍKU).....	59
C.11.ADCOFSH/L (ADC OFFSET CALIBRATION COEFFICIENT – POSUN A/D PŘEVODNÍKU).....	59
C.12.ADCSTAT (ADC STATUS REGISTER – STAVOVÝ REGISTR).....	59
C.13.CFG814 (ADUC814 CONFIGURATION REGISTER – KONFIGURACE).....	60
C.14.CFG831 (ADUC831 CONFIGURATION REGISTER – KONFIGURACE).....	60
C.15.CFG832 (ADUC832 CONFIGURATION REGISTER – KONFIGURACE).....	61
C.16.CFG834 (ADUC834 CONFIGURATION REGISTER – KONFIGURACE).....	61
C.17.CFG836 (ADUC836 CONFIGURATION REGISTER – KONFIGURACE).....	61
C.18.CFG841 (ADUC841 CONFIGURATION REGISTER – KONFIGURACE).....	62
C.19.CFG842 (ADUC842 CONFIGURATION REGISTER – KONFIGURACE).....	62
C.20.DACCON (DAC CONTROL – ŘÍDÍCÍ SLOVO D/A PŘEVODNÍKU).....	63
C.21.DACH/L (DAC DATA REG. – DATA DO D/A PŘEVODNÍKU).....	63
C.22.DACXH/L (DAC DATA REG. – DATA DO D/A PŘEVODNÍKU).....	64
C.23.DMAL/H/P (DMA ADDRESS POINTER – UKAZATEL DMA TABULKY).....	64
C.24.ECON (FLASH/EE DATA SPACE CONTROLS – ŘÍDÍCÍ SLOVO).....	64
C.25.EDARL/(H) (FLASH/EE DATA PAGE ADDRESS – STRÁNKA FLASH/EE).....	65
C.26.EDATA1 – 4 (FLASH/EE DATA).....	65
C.27.ETIM1 – 3 (FLASH/EE MEMORY TIMING).....	65
C.28.GN0H/M/(L) (PRIMARY ADC GAIN CALIBRATION – ZISK PRIMÁRNÍHO A/D).....	65
C.29.GN1H/L (AUXILIARY ADC GAIN CALIBRATION – ZISK POMOCNÉHO A/D).....	65
C.30.HOUR (HOURS TIME REGISTER – REGISTR HODIN).....	66
C.31.HTHSEC (HUNDRETHS SECONDS TIME REGISTER – REGISTR SETIN VTEŘINY).....	66
C.32.I2CADD (I2C ADDRESS REGISTER – I2C ADRESA).....	66
C.33.I2CADD1 (I2C ADDRESS REGISTER – I2C ADRESA).....	66
C.34.I2CADD2 (I2C ADDRESS REGISTER – I2C ADRESA).....	66
C.35.I2CADD3 (I2C ADDRESS REGISTER – I2C ADRESA).....	66
C.36.I2CCON (I2C CONTROL REGISTER – ŘÍDÍCÍ SLOVO I2C).....	67
C.37.I2CDAT (I2C DATA REGISTER – DATA PRO I2C PŘENOS).....	68
C.38.IE (INTERRUPT ENABLE – POVOLENÍ PŘERUŠENÍ).....	68
C.39.IE2 (SECONDARY INTERRUPT ENABLE – POVOLENÍ PŘERUŠENÍ 2).....	69
C.40.IEIP2 (SECONDARY INTERRUPT ENABLE AND PRIORITY – POVOLENÍ A PRIORITY PŘERUŠENÍ 2).....	69
C.41.INTVAL (USER TIME INTERVAL SELECT – UŽIVATELSKY NASTAVITELNÝ ČASOVAČ).....	69
C.42.IP (INTERRUPT PRIORITY – PRIORITY PŘERUŠENÍ).....	69
C.43.MIN (MINUTES TIME REGISTER – REGISTR MINUT).....	70
C.44.OF0H/M/(L) (PRIMARY ADC OFFSET CALIBRATION – POSUN PRIMÁRNÍHO A/D).....	70
C.45.OF1H/L (AUXILIARY ADC OFFSET CALIBRATION – POSUN POMOCNÉHO A/D).....	70
C.46.PLLCON (PLL CONTROL REGISTER – ŘÍDÍCÍ SLOVO PLL).....	70
C.47.PSMCON (POWER SUPPLY MONITOR – MONITOR NAPÁJENÍ).....	71
C.48.PWMCON (PWM CONTROL – ŘÍDÍCÍ SLOVO PWM).....	72
C.49.PWMxH/L (PWM DATA REGISTERS – DATA DO PWM PŘEVODNÍKU).....	72
C.50.SEC (SECONDS TIME REGISTER – REGISTR SEKUND).....	72

C.51.SPICON (SPI CONTROL – ŘÍDÍCI SLOVO SPI).....	73
C.52.SPIDAT (SPI DATA – DATA SPI KOMUNIKACE).....	73
C.53.TIMECON (TIC CONTROL – ŘÍDÍCI SLOVO TIC).....	74
C.54.WDCON (WATCHDOG TIMER – HLÍDACÍ ČASOVAČ).....	75
PŘÍLOHA D – INSTRUKČNÍ SADA 8051 (8052).....	76

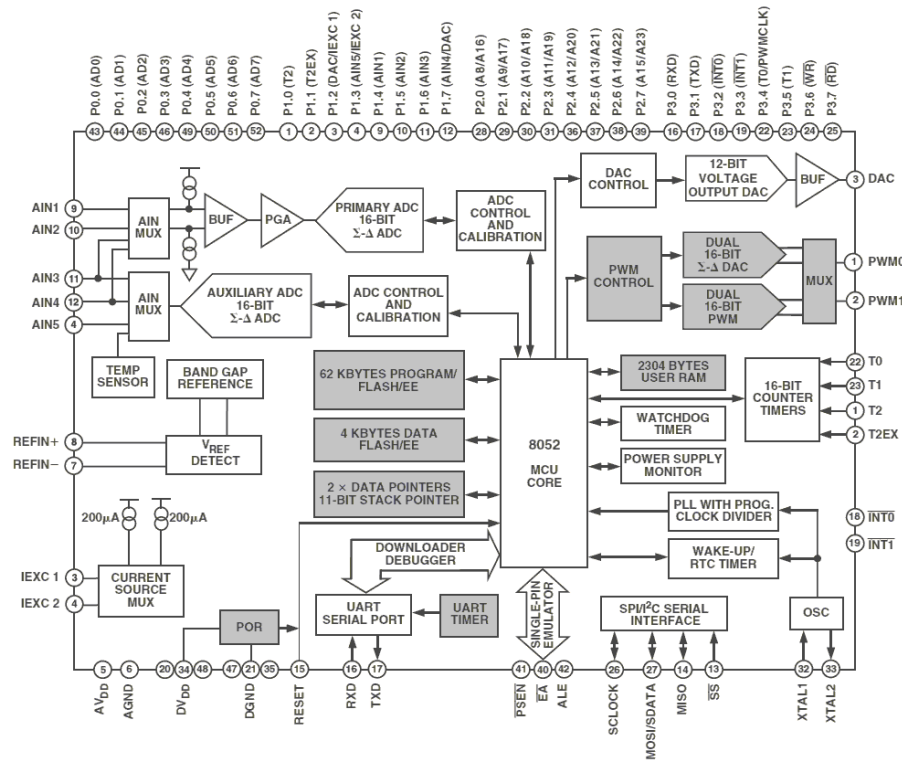
Příloha A – Blokové schéma ADuC8xx



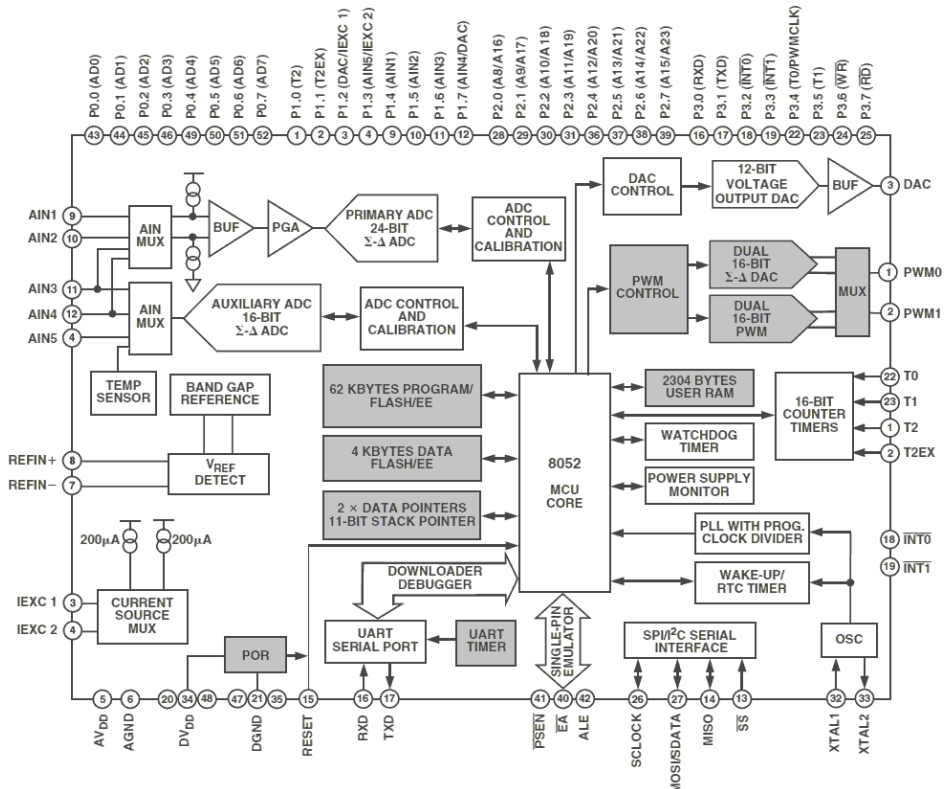
Obr. A.1 Blokové schéma ADuC812, ADuC831, ADuC832, ADuC841, ADuC842 a ADuC843 (šedě vybarvené bloky nejsou součástí ADuC812), blok DAC není osazen v ADuC843 a blok PLL jen v ADuC832, ADuC842 a ADuC843 [5], [9], [10], [13]



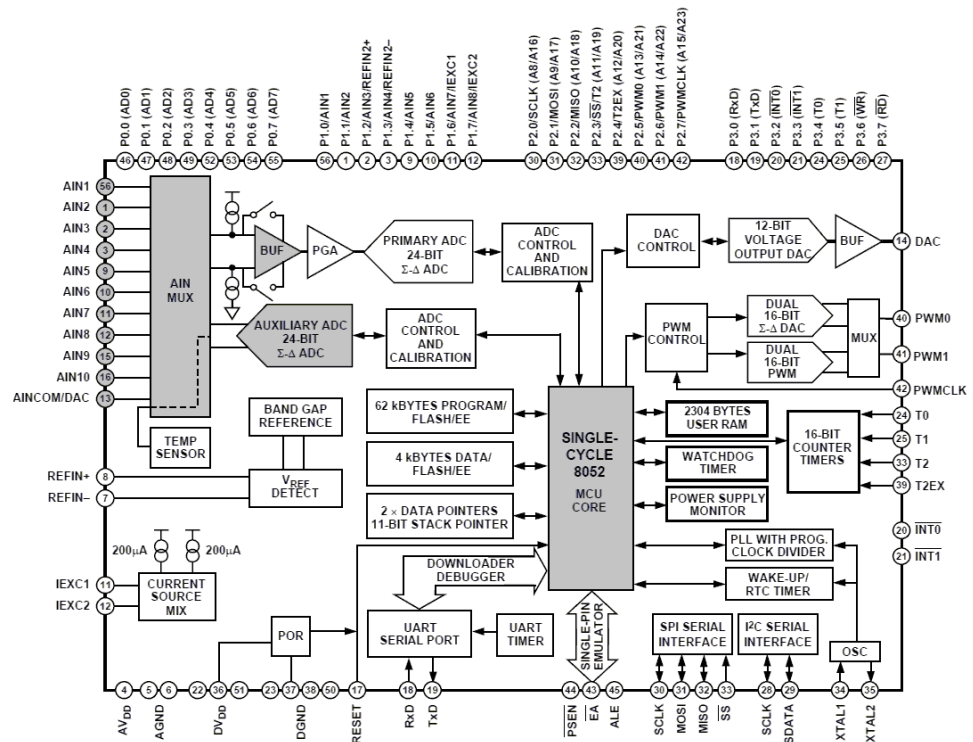
Obr. A.2: Blokové schéma ADuC814 [6]



Obr. A.3: Blokové schéma ADuC816, ADuC836 (šedě vybarvené bloky nejsou součástí ADuC816) [7], [12]

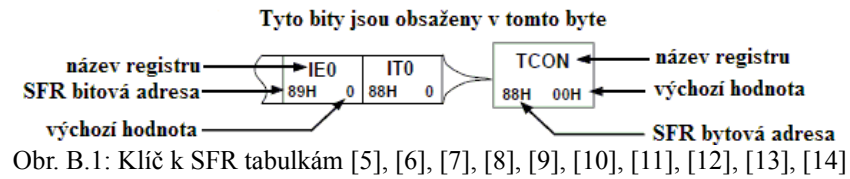


Obr. A.4: Blokové schéma ADuC824, ADuC834 (šedě vybarvené bloky nejsou součástí ADuC824) [8], [11]



Obr. A.5: Blokové schéma ADuC845, ADuC847 a ADuC848, Auxiliary (pomocný) A/D převodník jen v ADuC845, v ADuC848 16bitový převodník [14]

Příloha B – SPECIAL FUNCTION REGISTERS (SFR) oblast speciálních registrů



ISPI FFH 0	WCOL FEH 0	SPE FDH 0	SPIM FCH 0	CPOL FBH 0	CPHA FAH 0	SPR1 F9H 0	SPR0 F8H 0	BITS	SPICON ¹ F8H 00H	DAC0L F9H 00H	DAC0H FAH 00H	DAC1L FBH 00H	DAC1H FCH 00H	DACCON FDH 04H	RESERVED	NOT USED
F7H 0	F6H 0	F5H 0	F4H 0	F3H 0	F2H 0	F1H 0	F0H 0	BITS	B ¹ F0H 00H	ADCOFSL ² F1H 00H	ADCOFSH ² F2H 20H	ADCGAINL ² F3H 00H	ADCGAINH ² F4H 00H	ADCCON3 F5H 00H	RESERVED	SPIDAT F7H 00H
MDO EFH 0	MDE EEH 0	MCO EDH 0	MDI ECH 0	I2CM EBH 0	I2CRS EAH 0	I2CTX E9H 0	I2CI E8H 0	BITS	I2CCON ¹ E8H 00H	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	ADCCON1 EFH 20H
E7H 0	E6H 0	E5H 0	E4H 0	E3H 0	E2H 0	E1H 0	E0H 0	BITS	ACC ¹ E0H 00H	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
ADC1 DFH 0	DMA DEH 0	CCONV DDH 0	SCONV DCH 0	CS3 DBH 0	CS2 DAH 0	CS1 D9H 0	CS0 D8H 0	BITS	ADCCON2 ¹ D8H 00H	ADCCON2 ¹ D9H 00H	ADCCON2 ¹ DAH 00H	RESERVED	RESERVED	RESERVED	RESERVED	PSMCON DFH DEH
CY D7H 0	AC D6H 0	F0 D5H 0	RS1 D4H 0	RS0 D3H 0	OV D2H 0	FI D1H 0	P D0H 0	BITS	PSW ¹ D0H 00H	RESERVED	DMAL D2H 00H	DMAH D3H 00H	DMAP D4H 00H	RESERVED	RESERVED	RESERVED
TF2 CFH 0	EXF2 CEH 0	RCLK CDH 0	TCLK CCH 0	EXEN2 CBH 0	TR2 CAH 0	CNT2 C9H 0	CAP2 C8H 0	BITS	T2CON ¹ C8H 00H	RESERVED	RCAP2L CAH 00H	RCAP2H CBH 00H	TL2 CCH 00H	TH2 CDH 00H	RESERVED	RESERVED
PRE2 C7H 0	PRE1 C6H 0	PRE0 C5H 0	WDR1 C4H 0	WDR2 C3H 0	WDS C2H 0	WDE C1H 0	C0H 0	BITS	WDCON ¹ C0H 00H	NOT USED	NOT USED	NOT USED	ETIM3 C4H C9H	RESERVED	EDARL C6H 00H	RESERVED
PSI BFH 0	PADC BEH 0	PT2 BDH 0	PS BCH 0	PT1 BBH 0	PX1 BAH 0	PT0 B9H 0	PX0 B8H 0	BITS	IP ¹ B8H 00H	ECON B9H 00H	ETIM1 BAH 52H	ETIM2 BBH 04H	EDATA1 BCH 00H	EDATA2 BDH 00H	EDATA3 BEH 00H	EDATA4 BFH 00H
RD B7H 1	WR B6H 1	T1 B5H 1	T0 B4H 1	INT1 B3H 1	INT0 B2H 1	TxD B1H 1	RxD B0H 1	BITS	P3 ¹ B0H FFH	NOT USED	NOT USED	NOT USED	NOT USED	NOT USED	NOT USED	NOT USED
EA AFH 0	EADC AEH 0	ET2 ADH 0	ES ACH 0	ET1 ABH 0	EX1 AAH 0	ET0 A9H 0	EX0 A8H 0	BITS	IE ¹ A8H 00H	IE2 A9H 00H	NOT USED	NOT USED	NOT USED	NOT USED	NOT USED	NOT USED
A7H 1	A6H 1	A5H 1	A4H 1	A3H 1	A2H 1	A1H 1	A0H 1	BITS	P2 ¹ A0H FFH	NOT USED	NOT USED	NOT USED	NOT USED	NOT USED	NOT USED	NOT USED
SM0 9FH 0	SM1 9EH 0	SM2 9DH 0	REN 9CH 0	TB8 9BH 0	RB8 9AH 0	TI 99H 0	RI 98H 0	BITS	SCON ¹ 98H 00H	SBUF 99H 00H	I2CDAT 9AH 00H	I2CADD 9BH 55H	NOT USED	NOT USED	NOT USED	NOT USED
97H 1	96H 1	95H 1	94H 1	93H 1	92H 1	T2EX 91H 1	T2 90H 1	BITS	P1 ^{1,3} 90H FFH	NOT USED	NOT USED	NOT USED	NOT USED	NOT USED	NOT USED	NOT USED
TF1 8FH 0	TR1 8EH 0	TF0 8DH 0	TR0 8CH 0	IE1 8BH 0	IT1 8AH 0	IE0 89H 0	IT0 88H 0	BITS	TCON ¹ 88H 00H	TMOD 89H 00H	TL0 8AH 00H	TL1 8BH 00H	TH0 8CH 00H	TH1 8DH 00H	NOT USED	NOT USED
87H 1	86H 1	85H 1	84H 1	83H 1	82H 1	81H 1	80H 1	BITS	P0 ¹ 80H FFH	SP 81H 07H	DPL 82H 00H	DPH 83H 00H	DPP 84H 00H	RESERVED	RESERVED	PCON 87H 00H

Obr. B.2: Tabulka SFR ADuC812 [5]

ISPI FFH 0	WCOL FEH 0	SPE FDH 0	SPIM FCH 0	CPOL FBH 0	CPHA FAH 1	SPR1 F9H 0	SPR0 ⁵ F8H 0	BITS	SPICON ¹ F8H 04H	DAC0L F9H 00H	DAC0H FAH 00H	DAC1L FBH 00H	DAC1H FCH 00H	DACCON FDH 04H	RESERVED	RESERVED
F7H 0	F6H 0	F5H 0	F4H 0	F3H 0	F2H 0	F1H 0	F0H 0	BITS	B ¹ F0H 00H	ADCOFSL F1H 00H	ADCOFSH F2H 20H	ADCGAINL F3H 00H	ADCGAINH F4H 00H	ADCCON3 F5H 00H	RESERVED	SPIDAT F7H 00H
D1 EFH 0	D1EN EEH 0	D0 EDH 0	D0EN ECH 0	D0EN EBH 0	EAH 0	E9H 0	E8H 0	BITS	DCON ¹ E8H 00H	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	ADCCON1 EFH 00H
E7H 0	E6H 0	E5H 0	E4H 0	E3H 0	E2H 0	E1H 0	E0H 0	BITS	ACC ¹ E0H 00H	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
ADC1 DFH 0	ADCSP1 DEH 0	CCONV DDH 0	SCONV DCH 0	CS3 DBH 0	CS2 DAH 0	CS1 D9H 0	CS0 D8H 0	BITS	ADCCON2 ¹ D8H 00H	ADCCON2 D9H 00H	ADCCON2 DAH 00H	RESERVED	RESERVED	RESERVED	RESERVED	PSMCON DFH DEH
CY D7H 0	AC D6H 0	F0 D5H 0	RS1 D4H 0	RS0 D3H 0	OV D2H 0	FI D1H 0	P D0H 0	BITS	PSW ¹ D0H 00H	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	PLLCON D7H 53H
TF2 CFH 0	EXF2 CEH 0	RCLK CDH 0	TCLK CCH 0	EXEN2 CBH 0	TR2 CAH 0	CNT2 C9H 0	CAP2 C8H 0	BITS	T2CON ¹ C8H 00H	RESERVED	RCAP2L CAH 00H	RCAP2H CBH 00H	TL2 CCH 00H	TH2 CDH 00H	RESERVED	RESERVED
PRE3 C7H 0	PRE2 C6H 0	PRE1 C5H 0	PRE0 C4H 1	WDIR C3H 0	WDS C2H 0	WD C1H 0	WDWR C0H 0	BITS	WDCON ¹ C0H 10H	RESERVED	CHIPID C2H 0XH	RESERVED	NOT USED	RESERVED	EDARL C6H 00H	RESERVED
PSI BFH 0	PADC BEH 0	PT2 BDH 0	PS BCH 0	PT1 BBH 0	PX1 BAH 0	PT0 B9H 0	PX0 B8H 0	BITS	IP ¹ B8H 00H	ECON B9H 00H	ETIM1 BAH 00H	ETIM2 BBH 00H	EDATA1 BCH 00H	EDATA2 BDH 00H	EDATA3 BEH 00H	EDATA4 BFH 00H
RD B7H 1	WR B6H 1	T1 B5H 1	T0 B4H 1	INT1 B3H 1	INT0 B2H 1	TxD B1H 1	RxD B0H 1	BITS	P3 ¹ B0H FFH	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	NOT USED
EA AFH 0	EADC AEH 0	ET2 ADH 0	ES ACH 0	ET1 ABH 0	EX1 AAH 0	ET0 A9H 0	EX0 A8H 0	BITS	IE ¹ A8H 00H	IEIP2 A9H A0H	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
SM0 9FH 0	SM1 9EH 0	SM2 9DH 0	REN 9CH 0	TB8 9BH 0	RB8 9AH 0	T1 99H 0	R1 98H 0	BITS	SCON ¹ 98H 00H	SBUF 99H 00H	I2CDAT 9AH 00H	I2CADD 9BH 55H	CFG814 9CH 04H	NOT USED	NOT USED	NOT USED
97H 1	96H 1	95H 1	94H 1	93H 1	92H 1	T2EX 91H 1	T2 90H 1	BITS	P1 ^{1,2} 90H FFH	NOT USED	NOT USED	NOT USED	NOT USED	NOT USED	NOT USED	NOT USED
TF1 8FH 0	TR1 8EH 0	TF0 8DH 0	TR0 8CH 0	IE1 8BH 0	IT1 8AH 0	IE0 89H 0	IT0 88H 0	BITS	TCON ¹ 88H 00H	TMOD 89H 00H	TL0 8AH 00H	TL1 8BH 00H	TH0 8CH 00H	TH1 8DH 00H	RESERVED	RESERVED
87H 1	86H 1	85H 1	84H 1	83H 1	82H 1	81H 1	80H 1	BITS	NOT USED	SP 81H 07H	DPL 82H 00H	DPH 83H 00H	RESERVED	RESERVED	RESERVED	PCON 87H 00H

Obr. B.3: Tabulka SFR ADuC814 [6]

ISPI FFH 0	WCOL FEH 0	SPE FDH 0	SPIM FCH 0	CPOL FBH 0	CPHA FAH 1	SPR1 F9H 0	SPR0 F8H 0	BITS	SPICON	RESERVED	RESERVED	DACL FBH 00H	DACH FCH 00H	DACCON FDH 00H	RESERVED	RESERVED
F7H 0	F6H 0	F5H 0	F4H 0	F3H 0	F2H 0	F1H 0	F0H 0	BITS	B	RESERVED	RESERVED	NOT USED	RESERVED	RESERVED	RESERVED	SPIDAT F7H 00H
MDO EFH 0	MDE EEH 0	MCO EDH 0	MDI ECH 0	I2CM EBH 0	I2CRS EAH 0	I2CTX E9H 0	I2CI E8H 0	BITS	I2CCON E8H 00H	RESERVED	GNOM* EAH 55H	GN0H* EBH 53H	GN1L* ECH 9AH	GN1H* EDH 59H	RESERVED	RESERVED
E7H 0	E6H 0	E5H 0	E4H 0	E3H 0	E2H 0	E1H 0	E0H 0	BITS	ACC	RESERVED	OF0M* E2H 00H	OF0H* E3H 80H	OF1L* E4H 00H	OF1H* E5H 80H	RESERVED	RESERVED
RDY0 DFH 0	RDY1 DEH 0	CAL DDH 0	NOXREF DCH 0	ERR0 DBH 0	ERR1 DAH 0	D9H 0	D8H 0	BITS	ADCSTAT D8H 00H	RESERVED	ADC0M DAH 00H	ADC0H DBH 00H	ADC1L DCH 00H	ADC1H DDH 00H	RESERVED	PSMCON DFH DEH
CY D7H 0	AC D6H 0	F0 D5H 0	RSI D4H 0	RS0 D3H 0	OV D2H 0	FI D1H 0	P D0H 0	BITS	PSW D0H 00H	ADCMODE D1H 00H	ADC0CON D2H 07H	ADC1CON D3H 00H	SF D4H 45H	ICON D5H 00H	RESERVED	PLLCON D7H 03H
TF2 CFH 0	EXF2 CEH 0	RCLK CDH 0	TCLK CCH 0	EXEN2 CBH 0	TR2 CAH 0	CNT2 C9H 0	CAP2 C8H 0	BITS	T2CON C8H 00H	RESERVED	RCAP2L CAH 00H	RCAP2H CBH 00H	TL2 CCH 00H	TH2 CDH 00H	RESERVED	RESERVED
PRE3 C7H 0	PRE2 C6H 0	PRE1 C5H 0	PRE0 C4H 1	WDIR C3H 0	WDS C2H 0	WDE C1H 0	WDWR C0H 0	BITS	WDCON C0H 10H	RESERVED	CHIPID C2H 16H	RESERVED	RESERVED	RESERVED	EADRL C6H 00H	RESERVED
PSI BFH 0	PADC BEH 0	PT2 BDH 0	PS BCH 0	PT1 BBH 0	PX1 BAH 0	PT0 B9H 0	PX0 B8H 0	BITS	IP	ECON B9H 00H	RESERVED	RESERVED	EDATA1 BCH 00H	EDATA2 BDH 00H	EDATA3 BEH 00H	EDATA4 BFH 00H
RD B7H 1	WR B6H 1	T1 B5H 1	T0 B4H 1	INT1 B3H 1	INT0 B2H 1	TxD B1H 1	RxD B0H 1	BITS	P3	NOT USED	NOT USED	NOT USED	NOT USED	RESERVED	RESERVED	NOT USED
EA AFH 0	EADC AEH 0	ET2 ADH 0	ES ACH 0	ET1 ABH 0	EX1 AAH 0	ET0 A9H 0	EX0 A8H 0	BITS	IE	IEIP2 A9H A0H	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
A7H 1	A6H 1	A5H 1	A4H 1	A3H 1	A2H 1	A1H 1	A0H 1	BITS	P2	TIMECON A0H FFH	HTHSEC A1H 00H	SEC A2H 00H	MIN A3H 00H	HOUR A4H 00H	INTVAL A5H 00H	NOT USED
SM0 9FH 0	SM1 9EH 0	SM2 9DH 0	REN 9CH 0	TB8 9BH 0	RB8 9AH 0	T1 99H 0	R1 98H 0	BITS	SCON	SBUF 99H 00H	I2CDAT 9AH 00H	I2CDAT 9BH 00H	NOT USED	NOT USED	NOT USED	NOT USED
97H 1	96H 1	95H 1	94H 1	93H 1	92H 1	T2EX 91H 1	T2 90H 1	BITS	P1	NOT USED	NOT USED	NOT USED	NOT USED	NOT USED	NOT USED	NOT USED
TF1 8FH 0	TR1 8EH 0	TF0 8DH 0	TR0 8CH 0	IE1 8BH 0	IT1 8AH 0	IE0 89H 0	IT0 88H 0	BITS	TCON	TMOD 89H 00H	TL0 8AH 00H	TL1 8BH 00H	TH0 8CH 00H	TH1 8DH 00H	RESERVED	RESERVED
87H 1	86H 1	85H 1	84H 1	83H 1	82H 1	81H 1	80H 1	BITS	P0	SP 81H 07H	DPL 82H 00H	DPH 83H 00H	DPP 84H 00H	RESERVED	RESERVED	PCON 87H 00H

Obr. B.4: Tabulka SFR ADuC816 [7]

ISPI FFH 0	WCOL FEH 0	SPE FDH 0	SPIM FCH 0	CPOL FBH 0	CPHA FAH 1	SPR1 F9H 0	SPR0 F8H 0	BITS	SPICON F8H 04H	RESERVED	RESERVED	DACL FBH 00H	DACH FCH 00H	DACCON FDH 00H	RESERVED	RESERVED
F7H 0	F6H 0	F5H 0	F4H 0	F3H 0	F2H 0	F1H 0	F0H 0	BITS	B F0H 00H	RESERVED	RESERVED	NOT USED	RESERVED	RESERVED	RESERVED	SPIDAT F7H 00H
MDO EFH 0	MDE EEH 0	MCO EDH 0	MDI ECH 0	I2CM EBH 0	I2CRS EAH 0	I2CTX E9H 0	I2CI E8H 0	BITS	I2CCON E8H 00H	GN0L* E9H 55H	GN0M* EAH 55H	GN0H* EBH 53H	GN1L* ECH 9AH	GN1H* EDH 59H	RESERVED	RESERVED
E7H 0	E6H 0	E5H 0	E4H 0	E3H 0	E2H 0	E1H 0	E0H 0	BITS	ACC E0H 00H	OF0L* E1H 00H	OF0M* E2H 00H	OF0H* E3H 80H	OF1L* E4H 00H	OF1H* E5H 80H	RESERVED	RESERVED
RDY0 DFH 0	RDY1 DEH 0	CAL DDH 0	NOXREF DCH 0	ERR0 DBH 0	ERR1 DAH 0	D9H 0	D8H 0	BITS	ADCSTAT D8H 00H	ADC0L D9H 00H	ADC0M DAH 00H	ADC0H DBH 00H	ADC1L DCH 00H	ADC1H DDH 00H	RESERVED	PSMCON DFH DEH
CY D7H 0	AC D6H 0	F0 D5H 0	RSI D4H 0	RS0 D3H 0	OV D2H 0	FI D1H 0	P D0H 0	BITS	PSW D0H 00H	ADCMODE D1H 00H	ADC0CON D2H 07H	ADC1CON D3H 00H	SF D4H 45H	ICON D5H 00H	RESERVED	PLLCON D7H 03H
TF2 CFH 0	EXF2 CEH 0	RCLK CDH 0	TCLK CCH 0	EXEN2 CBH 0	TR2 CAH 0	CNT2 C9H 0	CAP2 C8H 0	BITS	T2CON C8H 00H	RESERVED	RCAP2L CAH 00H	RCAP2H CBH 00H	TL2 CCH 00H	TH2 CDH 00H	RESERVED	RESERVED
PRE3 C7H 0	PRE2 C6H 0	PRE1 C5H 0	PRE0 C4H 1	WDIR C3H 0	WDS C2H 0	WDE C1H 0	WDWR C0H 0	BITS	WDCON C0H 10H	RESERVED	CHIPID C2H 06H	RESERVED	RESERVED	RESERVED	EADRL C6H 00H	RESERVED
BFH 0	PADC BEH 0	PT2 BDH 0	PS BCH 0	PT1 BBH 0	PX1 BAH 0	PT0 B9H 0	PX0 B8H 0	BITS	IP B8H 00H	ECON B9H 00H	RESERVED	RESERVED	EDATA1 BCH 00H	EDATA2 BDH 00H	EDATA3 BEH 00H	EDATA4 BFH 00H
RD B7H 1	WR B6H 1	T1 B5H 1	T0 B4H 1	INT1 B3H 1	INT0 B2H 1	TxD B1H 1	RxD B0H 1	BITS	P3 B0H FFH	NOT USED	NOT USED	NOT USED	NOT USED	RESERVED	RESERVED	NOT USED
EA AFH 0	EADC AEH 0	ET2 ADH 0	ES ACH 0	ET1 ABH 0	EX1 AAH 0	ET0 A9H 0	EX0 A8H 0	BITS	IE A8H 00H	IEIP2 A9H A0H	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
A7H 1	A6H 1	A5H 1	A4H 1	A3H 1	A2H 1	A1H 1	A0H 1	BITS	P2 A0H FFH	TIMECON A1H 00H	HTHSEC A2H 00H	SEC A3H 00H	MIN A4H 00H	HOUR A5H 00H	INTVAL A6H 00H	NOT USED
SM0 9FH 0	SM1 9EH 0	SM2 9DH 0	REN 9CH 0	TB8 9BH 0	RB8 9AH 0	T1 99H 0	R1 98H 0	BITS	SCON 98H 00H	SBUF 99H 00H	I2CDAT 9AH 00H	I2CDAT 9AH 00H	NOT USED	NOT USED	NOT USED	NOT USED
97H 1	96H 1	95H 1	94H 1	93H 1	92H 1	T2EX 91H 1	T2 90H 1	BITS	P1 90H FFH	NOT USED	NOT USED	NOT USED	NOT USED	NOT USED	NOT USED	NOT USED
TF1 8FH 0	TR1 8EH 0	TF0 8DH 0	TR0 8CH 0	IE1 8BH 0	IT1 8AH 0	IE0 89H 0	IT0 88H 0	BITS	TCON 88H 00H	TMOD 89H 00H	TL0 8AH 00H	TL1 8BH 00H	TH0 8CH 00H	TH1 8DH 00H	RESERVED	RESERVED
87H 1	86H 1	85H 1	84H 1	83H 1	82H 1	81H 1	80H 1	BITS	P0 80H FFH	SP 81H 07H	DPL 82H 00H	DPH 83H 00H	DPP 84H 00H	RESERVED	RESERVED	PCON 87H 00H

Obr. B.5: Tabulka SFR ADuC824 [8]

ISPI FFH 0	WCOL FEH 0	SPE FDH 0	SPIM FCH 0	CPOL FBH 0	CPHA FAH 1	SPR1 F9H 0	SPR0 F8H 0	BITS	SPICON ¹ F8H 04H	DAC0L F9H 00H	DAC0H FAH 00H	DAC1L FBH 00H	DAC1H FCH 00H	DACCON FDH 04H	RESERVED	RESERVED
F7H 0	F6H 0	F5H 0	F4H 0	F3H 0	F2H 0	F1H 0	F0H 0	BITS	B ¹ F0H 00H	ADCOFSL ³ F1H 00H	ADCOFSH ³ F2H 20H	ADCGAINL ³ F3H 00H	ADCGAINH ³ F4H 00H	ADCCON3 F5H 00H	RESERVED	SPIDAT F7H 00H
MDO EFH 0	MDE EEH 0	MCO EDH 0	MDI ECH 0	I2CM EBH 0	I2CRS EAH 0	I2CTX E9H 0	I2CI E8H 0	BITS	I2CCON ¹ E8H 00H	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	ADCCON1 EFH 00H
E7H 0	E6H 0	E5H 0	E4H 0	E3H 0	E2H 0	E1H 0	E0H 0	BITS	ACC ¹ E0H 00H	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
ADC1 DFH 0	DMA DEH 0	CCONV DDH 0	SCONV DCH 0	CS3 DBH 0	CS2 DAH 0	CS1 D9H 0	CS0 D8H 0	BITS	ADCCON2 ¹ D8H 00H	ADCDATL D9H 00H	ADCDATAH DAH 00H	RESERVED	RESERVED	RESERVED	RESERVED	PSMCON DFH DEH
CY D7H 0	AC D6H 0	F0 D5H 0	RS1 D4H 0	RS0 D3H 0	OV D2H 0	FI D1H 0	P D0H 0	BITS	PSW ¹ D0H 00H	RESERVED	DMAL D2H 00H	DMAH D3H 00H	DMAP D4H 00H	RESERVED	RESERVED	RESERVED
TF2 CFH 0	EXF2 CEH 0	RCLK CDH 0	TCLK CCH 0	EXEN2 CBH 0	TR2 CAH 0	CNT2 C9H 0	CAP2 C8H 0	BITS	T2CON ¹ C8H 00H	RESERVED	RCAP2L CAH 00H	RCAP2H CBH 00H	TL2 CCH 00H	TH2 CDH 00H	RESERVED	RESERVED
PRE3 C7H 0	PRE2 C6H 0	PRE1 C5H 0	PRE0 C4H 1	WDIR C3H 0	WDS C2H 0	WDE C1H 0	WDWR C0H 0	BITS	WDCON ¹ C0H 10H	RESERVED	CHIPID C2H 3XH	RESERVED	RESERVED	RESERVED	EDARL C6H 00H	EDARH C7H 00H
PSI BFH 0	PADC BEH 0	PT2 BDH 0	PS BCH 0	PT1 BBH 0	PX1 BAH 0	PT0 B9H 0	PX0 B8H 0	BITS	IP ¹ B8H 00H	ECON B9H 00H	RESERVED	RESERVED	EDATA1 BCH 00H	EDATA2 BDH 00H	EDATA3 BEH 00H	EDATA4 BFH 00H
RD B7H 1	WR B6H 1	T1 B5H 1	T0 B4H 1	INT1 B3H 1	INT0 B2H 1	TxD B1H 1	RxD B0H 1	BITS	P3 ¹ B0H FFH	PWM0L B1H 00H	PWM0H B2H 00H	PWM1L B3H 00H	PWM1H B4H 00H	NOT USED	NOT USED	SPH B7H 00H
EA AFH 0	EADC AEH 0	ET2 ADH 0	ES ACH 0	ET1 ABH 0	EX1 AAH 0	ET0 A9H 0	EX0 A8H 0	BITS	IE ¹ A8H 00H	IEIP2 A9H A0H	RESERVED	RESERVED	RESERVED	RESERVED	PWMCON AEH 00H	CFG831 ⁴ AFH 10H
A7H 1	A6H 1	A5H 1	A4H 1	A3H 1	A2H 1	A1H 1	A0H 1	BITS	P2 ¹ A0H FFH	TIMECON A1H 00H	HTHSEC A2H 00H	SEC A3H 00H	MIN A4H 00H	HOUR A5H 00H	INTVAL A6H 00H	DPCON A7H 00H
SM0 9FH 0	SM1 9EH 0	SM2 9DH 0	REN 9CH 0	TB8 9BH 0	RB8 9AH 0	T1 99H 0	R1 98H 0	BITS	SCON ¹ 98H 00H	SBUF 99H 00H	I2CDAT 9AH 00H	I2CADD 9BH 55H	NOT USED	T3FD 9DH 00H	T3CON 9EH 00H	NOT USED
97H 1	96H 1	95H 1	94H 1	93H 1	92H 1	T2EX 91H 1	T2 90H 1	BITS	P1 ^{1,2} 90H FFH	NOT USED	NOT USED	NOT USED	NOT USED	NOT USED	NOT USED	NOT USED
TF1 8FH 0	TR1 8EH 0	TF0 8DH 0	TR0 8CH 0	IE1 8BH 0	IT1 8AH 0	IE0 89H 0	IT0 88H 0	BITS	TCON ¹ 88H 00H	TMOD 89H 00H	TL0 8AH 00H	TL1 8BH 00H	TH0 8CH 00H	TH1 8DH 00H	RESERVED	RESERVED
87H 1	86H 1	85H 1	84H 1	83H 1	82H 1	81H 1	80H 1	BITS	P0 ¹ 80H FFH	SP 81H 07H	DPL 82H 00H	DPH 83H 00H	DPP 84H 00H	RESERVED	RESERVED	PCON 87H 00H

Obr. B.6: Tabulka SFR ADuC831 [9]

ISPI FFH 0	WCOL FEH 0	SPE FDH 0	SPIM FCH 0	CPOL FBH 0	CPHA FAH 1	SPR1 F9H 0	SPR0 F8H 0	BITS	SPICON ¹ F8H 04H	DAC0L F9H 00H	DAC0H FAH 00H	DAC1L FBH 00H	DAC1H FCH 00H	DACCON FDH 04H	RESERVED	RESERVED
F7H 0	F6H 0	F5H 0	F4H 0	F3H 0	F2H 0	F1H 0	F0H 0	BITS	B ¹ F0H 00H	ADCCFSL ³ F1H 00H	ADCCFSL ³ F2H 20H	ADCGANL ³ F3H 00H	ADCGANH ³ F4H 00H	ADCCON3 F5H 00H	RESERVED	SPIDAT F7H 00H
MDO EFH 0	MDE EEH 0	MCO EDH 0	MDI ECH 0	I2CM EBH 0	I2CRS EAH 0	I2CTX E9H 0	I2CI E8H 0	BITS	I2CCON ¹ E8H 00H	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	ADCCON1 EFH 00H
E7H 0	E6H 0	E5H 0	E4H 0	E3H 0	E2H 0	E1H 0	E0H 0	BITS	ACC ¹ E0H 00H	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
ADCI DFH 0	DMA DEH 0	CCONV DDH 0	SCONV DCH 0	CS3 DBH 0	CS2 DAH 0	CS1 D9H 0	CS0 D8H 0	BITS	ADCCON2 ¹ D8H 00H	ADCDATAL D9H 00H	ADCDATAH DAH 00H	RESERVED	RESERVED	RESERVED	RESERVED	PSMCON DFH DEH
CY D7H 0	AC D6H 0	F0 D5H 0	RS1 D4H 0	RS0 D3H 0	OV D2H 0	F1 D1H 0	P D0H 0	BITS	PSW ¹ D0H 00H	RESERVED	DMAL D2H 00H	DMAH D3H 00H	DMAP D4H 00H	RESERVED	RESERVED	PLLCON D7H 53H
TF2 CFH 0	EXF2 CEH 0	RCLK CDH 0	TCLK CCH 0	EXEN2 CBH 0	TR2 CAH 0	CNT2 C9H 0	CAP2 C8H 0	BITS	T2CON ¹ C8H 00H	RESERVED	RCAP2L CAH 00H	RCAP2H CBH 00H	TL2 CCH 00H	TH2 CDH 00H	RESERVED	RESERVED
PRE3 C7H 0	PRE2 C6H 0	PRE1 C5H 0	PRE0 C4H 1	WDIR C3H 0	WDS C2H 0	WDE C1H 0	WDWR C0H 0	BITS	WDCON ¹ C0H 10H	RESERVED	CHIPID C2H 2XH	RESERVED	RESERVED	RESERVED	EADRL C6H 00H	EADRH C7H 00H
BFH 0	PADC BEH 0	PT2 BDH 0	PS BCH 0	PT1 BBH 0	PX1 BAH 0	PT0 B9H 0	PX0 B8H 0	BITS	IP ¹ B8H 00H	ECON B9H 00H	RESERVED	RESERVED	EDATA1 BCH 00H	EDATA2 BDH 00H	EDATA3 BEH 00H	EDATA4 BFH 00H
RD B7H 1	WR B6H 1	T1 B5H 1	T0 B4H 1	INT1 B3H 1	INT0 B2H 1	TxD B1H 1	RxD B0H 1	BITS	P3 ¹ B0H FFH	PWM0L B1H 00H	PWM0H B2H 00H	PWM1L B3H 00H	PWM1H B4H 00H	NOT USED	NOT USED	SPH B7H 00H
EA AFH 0	EADC AEH 0	ET2 ADH 0	ES ACH 0	ET1 ABH 0	EX1 AAH 0	ET0 A9H 0	EX0 A8H 0	BITS	IE ¹ A8H 00H	IEIP2 A9H A0H	RESERVED	RESERVED	RESERVED	RESERVED	PWMCON AEH 00H	CFG832 AFH 00H
A7H 1	A6H 1	A5H 1	A4H 1	A3H 1	A2H 1	A1H 1	A0H 1	BITS	P2 ¹ A0H FFH	TIMECON A1H 00H	HTHSEC A2H 00H	SEC A3H 00H	MIN A4H 00H	hour A5H 00H	INTVAL A6H 00H	DPCON A7H 00H
SM0 9FH 0	SM1 9EH 0	SM2 9DH 0	REN 9CH 0	TB8 9BH 0	RB8 9AH 0	T1 99H 0	R1 98H 0	BITS	SCON ¹ 98H 00H	SBUF 99H 00H	I2CDAT 9AH 00H	I2CADD 9BH 55H	NOT USED	T3FD 9DH 00H	T3CON 9EH 00H	NOT USED
97H 1	96H 1	95H 1	94H 1	93H 1	92H 1	T2EX 91H 1	T2 90H 1	BITS	P1 ^{1,2} 90H FFH	NOT USED	NOT USED	NOT USED	NOT USED	NOT USED	NOT USED	NOT USED
TF1 8FH 0	TR1 8EH 0	TF0 8DH 0	TR0 8CH 0	IE1 8BH 0	IT1 8AH 0	IE0 89H 0	IT0 88H 0	BITS	TCON ¹ 88H 00H	TMOD 89H 00H	TL0 8AH 00H	TL1 8BH 00H	TH0 8CH 00H	TH1 8DH 00H	RESERVED	RESERVED
87H 1	86H 1	85H 1	84H 1	83H 1	82H 1	81H 1	80H 1	BITS	P0 ¹ 80H FFH	SP 81H 07H	DPL 82H 00H	DPH 83H 00H	DPP 84H 00H	RESERVED	RESERVED	PCON 87H 00H

Obr. B.7: Tabulka SFR ADuC832 [10]

ISPI FFH 0	WCOL FEH 0	SPE FDH 0	SPIM FCH 0	CPOL FBH 0	CPHA FAH 1	SPR1 F9H 0	SPR0 F8H 0	BITS	SPICON	RESERVED	RESERVED	DACL FBH 00H	DACH FCH 00H	DACCON FDH 00H	RESERVED	RESERVED	
F7H 0	F6H 0	F5H 0	F4H 0	F3H 0	F2H 0	F1H 0	F0H 0	BITS	B	RESERVED	RESERVED	NOT USED	RESERVED	RESERVED	RESERVED	SPIDAT F7H 00H	
MDO EFH 0	MDE EEH 0	MCO EDH 0	MDI ECH 0	I2CM EBH 0	I2CRS EAH 0	I2CTX E9H 0	I2CI E8H 0	BITS	I2CCON	GN0L ¹ E8H 00H	GN0M ¹ E9H 55H	GN0H ¹ EAH 55H	GN1L ¹ EBH 53H	GN1H ¹ ECH 9AH	EDH 59H	RESERVED	RESERVED
E7H 0	E6H 0	E5H 0	E4H 0	E3H 0	E2H 0	E1H 0	E0H 0	BITS	ACC	OF0L E0H 00H	OF0M E1H 00H	OF0H E2H 00H	OF1L E3H 80H	OF1H E4H 00H	ESH 80H	RESERVED	RESERVED
RDY0 DFH 0	RDY1 DEH 0	CAL DDH 0	NOXREF DCH 0	ERR0 DBH 0	ERR1 DAH 0	D9H 0	D8H 0	BITS	ADCSTAT	ADCOL D8H 00H	ADCOM D9H 00H	ADCOH DAH 00H	ADC1L DBH 00H	ADC1H DCH 00H	DDH 00H	RESERVED	PSMCON DFH DEH
CY D7H 0	AC D6H 0	F0 D5H 0	RSI D4H 0	RS0 D3H 0	OV D2H 0	F1 D1H 0	P D0H 0	BITS	PSW	ADCMODE D0H 00H	ADC0CON D1H 00H	ADC1CON D2H 07H	SF D3H 00H	ICON D4H 45H	D5H 00H	RESERVED	PLLCON D7H 03H
TF2 CFH 0	EXF2 CEH 0	RCLK CDH 0	TCLK CCH 0	EXEN2 CBH 0	TR2 CAH 0	CNT2 C9H 0	CAP2 C8H 0	BITS	T2CON	RESERVED	RCAP2L CAH 00H	RCAP2H CBH 00H	TL2 CCH 00H	TH2 CDH 00H	RESERVED	RESERVED	
PRE3 C7H 0	PRE2 C6H 0	PRE1 C5H 0	PRE0 C4H 1	WDIR C3H 0	WDS C2H 0	WDE C1H 0	WDWR C0H 0	BITS	WDCON	RESERVED	CHIPID C2H 2xH	RESERVED	RESERVED	RESERVED	EADRL C6H 00H	EADRH C7H 00H	
BFH 0	PADC BEH 0	PT2 BDH 0	PS BCH 0	PT1 BBH 0	PX1 BAH 0	PT0 B9H 0	PX0 B8H 0	BITS	IP	ECON B8H 00H	RESERVED	RESERVED	EDATA1 BCH 00H	EDATA2 BDH 00H	EDATA3 BEH 00H	EDATA4 BFH 00H	
RD B7H 1	WR B6H 1	T1 B5H 1	T0 B4H 1	INT1 B3H 1	INT0 B2H 1	TxD B1H 1	RxD B0H 1	BITS	P3	PWM0L B0H FFH	PWM0H B1H 00H	PWM1L B2H 00H	PWM1H B3H 00H	B4H 00H	RESERVED	RESERVED	SPH B7H 00H
EA AFH 0	EADC AEH 0	ET2 ADH 0	ES ACH 0	ET1 ABH 0	EX1 AAH 0	ET0 A9H 0	EX0 A8H 0	BITS	IE	IEIP2 A8H 00H	RESERVED	RESERVED	RESERVED	RESERVED	PWMCON AEH 00H	CFG834 AFH 00H	
A7H 1	A6H 1	A5H 1	A4H 1	A3H 1	A2H 1	A1H 1	A0H 1	BITS	P2	TIMECON A0H FFH	HTHSEC ² A1H 00H	SEC ² A2H 00H	MIN ² A3H 00H	hour ² A4H 00H	A5H 00H	A6H 00H	INTVAL A7H 00H
SM0 9FH 0	SM1 9EH 0	SM2 9DH 0	REN 9CH 0	TB8 9BH 0	RB8 9AH 0	T1 99H 0	R1 98H 0	BITS	SCON	SBUF 98H 00H	RESERVED	RESERVED	NOT USED	T3FD 9DH 00H	T3CON 9EH 00H	RESERVED	
97H 1	96H 1	95H 1	94H 1	93H 1	92H 1	T2EX 91H 1	T2 90H 1	BITS	P1	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	
TF1 8FH 0	TR1 8EH 0	TF0 8DH 0	TR0 8CH 0	IE1 8BH 0	IT1 8AH 0	IE0 89H 0	IT0 88H 0	BITS	TCON	TMOD 88H 00H	TL0 89H 00H	TL1 8AH 00H	TH0 8BH 00H	TH1 8CH 00H	8DH 00H	RESERVED	RESERVED
87H 1	86H 1	85H 1	84H 1	83H 1	82H 1	81H 1	80H 1	BITS	P0	SP 80H FFH	DPL 81H 07H	DPH 82H 00H	DPH 83H 00H	DPP 84H 00H	RESERVED	RESERVED	PCON 87H 00H

Obr. B.8: Tabulka SFR ADuC834 [11]

ISPI FFH 0	WCOL FEH 0	SPE FDH 0	SPIM FCH 0	CPOL FBH 0	CPHA FAH 1	SPR1 F9H 0	SPR0 F8H 0	BITS	SPICON F8H 04H	RESERVED	RESERVED	DACL FBH 00H	DACH FCH 00H	DACCON FDH 00H	RESERVED	RESERVED
F7H 0	F6H 0	F5H 0	F4H 0	F3H 0	F2H 0	F1H 0	F0H 0	BITS	B F0H 00H	RESERVED	RESERVED	NOT USED	RESERVED	RESERVED	RESERVED	SPIDAT F7H 00H
MDO EFH 0	MDE EEH 0	MCO EDH 0	MDI ECH 0	I2CM EBH 0	I2CRS EAH 0	I2CTX E9H 0	I2CI E8H 0	BITS	I2CCON E8H 00H	RESERVED	GNOM ¹ EAH 55H	GNOH ¹ EBH 53H	GN1L ¹ ECH 9AH	GN1H ¹ EDH 59H	RESERVED	RESERVED
E7H 0	E6H 0	E5H 0	E4H 0	E3H 0	E2H 0	E1H 0	E0H 0	BITS	ACC E0H 00H	RESERVED	OF0M E2H 00H	OF0H E3H 80H	OF1L E4H 00H	OF1H E5H 80H	RESERVED	RESERVED
RDY0 DFH 0	RDY1 DEH 0	CAL DDH 0	NOXREF DCH 0	ERR0 DBH 0	ERR1 DAH 0	D9H 0	D8H 0	BITS	ADCSTAT D8H 00H	RESERVED	ADCOM DAH 00H	ADC0H DBH 00H	ADC1L DCH 00H	ADC1H DDH 00H	RESERVED	PSMCON DFH DEH
CY D7H 0	AC D6H 0	F0 D5H 0	RSI D4H 0	RS0 D3H 0	OV D2H 0	FI D1H 0	P D0H 0	BITS	PSW D0H 00H	ADCMODE D1H 00H	ADCOCON D2H 07H	ADC1CON D3H 00H	SF D4H 45H	ICON D5H 00H	RESERVED	PLLCON D7H 03H
TF2 CFH 0	EXF2 CEH 0	RCLK CDH 0	TCLK CCH 0	EXEN2 CBH 0	TR2 CAH 0	CNT2 C9H 0	CAP2 C8H 0	BITS	T2CON C8H 00H	RESERVED	RCAP2L CAH 00H	RCAP2H CBH 00H	TL2 CCH 00H	TH2 CDH 00H	RESERVED	RESERVED
PRE3 C7H 0	PRE2 C6H 0	PRE1 C5H 0	PRE0 C4H 1	WDIR C3H 0	WDS C2H 0	WDE C1H 0	WDWR C0H 0	BITS	WDCON C0H 10H	RESERVED	CHIPID C2H 2xH	RESERVED	RESERVED	RESERVED	EADRL C6H 00H	EADRH C7H 00H
BFH 0	PADC BEH 0	PT2 BDH 0	PS BCH 0	PT1 BBH 0	PX1 BAH 0	PT0 B9H 0	PX0 B8H 0	BITS	IP B8H 00H	ECON B9H 00H	RESERVED	RESERVED	EDATA1 BCH 00H	EDATA2 BDH 00H	EDATA3 BEH 00H	EDATA4 BFH 00H
RD B7H 1	WR B6H 1	T1 B5H 1	T0 B4H 1	INT1 B3H 1	INT0 B2H 1	TxD B1H 1	RxD B0H 1	BITS	P3 B0H FFH	PWM0L B1H 00H	PWM0H B2H 00H	PWM1L B3H 00H	PWM1H B4H 00H	RESERVED	RESERVED	SPH B7H 00H
EA AFH 0	EADC AEH 0	ET2 ADH 0	ES ACH 0	ET1 ABH 0	EX1 AAH 0	ET0 A9H 0	EX0 A8H 0	BITS	IE A8H 00H	IEIP2 A9H A0H	RESERVED	RESERVED	RESERVED	RESERVED	PWMCON AEH 00H	CFG836 AFH 00H
A7H 1	A6H 1	A5H 1	A4H 1	A3H 1	A2H 1	A1H 1	A0H 1	BITS	P2 A0H FFH	TIMECON A1H 00H	HTHSEC ² A2H 00H	SEC ² A3H 00H	MIN ² A4H 00H	HOURL ² A5H 00H	INTVAL A6H 00H	DPCON A7H 00H
SM0 9FH 0	SM1 9EH 0	SM2 9DH 0	REN 9CH 0	TB8 9BH 0	RB8 9AH 0	T1 99H 0	R1 98H 0	BITS	SCON 98H 00H	SBUF 99H 00H	RESERVED	RESERVED	NOT USED	T3FD 9DH 00H	T3CON 9EH 00H	RESERVED
97H 1	96H 1	95H 1	94H 1	93H 1	92H 1	T2EX 91H 1	T2 90H 1	BITS	P1 90H FFH	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
TF1 8FH 0	TR1 8EH 0	TF0 8DH 0	TR0 8CH 0	IE1 8BH 0	IT1 8AH 0	IE0 89H 0	IT0 88H 0	BITS	TCON 88H 00H	TMOD 89H 00H	TL0 8AH 00H	TL1 8BH 00H	TH0 8CH 00H	TH1 8DH 00H	RESERVED	RESERVED
87H 1	86H 1	85H 1	84H 1	83H 1	82H 1	81H 1	80H 1	BITS	P0 80H FFH	SP 81H 07H	DPL 82H 00H	DPH 83H 00H	DPP 84H 00H	RESERVED	RESERVED	PCON 87H 00H

Obr. B.9: Tabulka SFR ADuC836 [12]

ISPI FFH 0	WCOL FEH 0	SPE FDH 0	SPIM FCH 0	CPOL FBH 0	CPHA FAH 1	SPR1 F9H 0	SPR0 F8H 0	BITS	SPICON ¹ F8H 04H	DACL0L F9H 00H	DACL0H FAH 00H	DACL1L FBH 00H	DACL1H FCH 00H	DACCON FDH 04H	RESERVED	RESERVED
F7H 0	F6H 0	F5H 0	F4H 0	F3H 0	F2H 0	F1H 0	F0H 0	BITS	B ¹ F0H 00H	ADCOFSL ³ F1H 00H	ADCOFSH ³ F2H 20H	ADCGAINL ³ F3H 00H	ADCGAINH ³ F4H 00H	ADCCON3 F5H 00H	RESERVED	SPIDAT F7H 00H
I2CSIMDO EFH 0	I2CGCMDDE EEH 0	I2C101MCO EDH 0	I2C100MDI ECH 0	I2CM EBH 0	I2CRS EAH 0	I2CTX E9H 0	I2CI E8H 0	BITS	I2CCON ¹ E8H 00H	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	ADCCON1 EFH 40H
E7H 0	E6H 0	E5H 0	E4H 0	E3H 0	E2H 0	E1H 0	E0H 0	BITS	ACC ¹ E0H 00H	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
ADC1 DFH 0	DMA DEH 0	CCONV DDH 0	SCONV DCH 0	CS3 DBH 0	CS2 DAH 0	CS1 D9H 0	CS0 D8H 0	BITS	ADCCON2 ¹ D8H 00H	ADCCDATA1 D9H 00H	ADCCDATAH DAH 00H	RESERVED	RESERVED	RESERVED	RESERVED	PSMCON DFH DEH
CY D7H 0	AC D6H 0	F0 D5H 0	RS1 D4H 0	RS0 D3H 0	OV D2H 0	FI D1H 0	P D0H 0	BITS	PSW ¹ D0H 00H	RESERVED	DMAL D2H 00H	DMAH D3H 00H	DMAP D4H 00H	RESERVED	RESERVED	PLLCON D7H 53H
TF2 CFH 0	EXF2 CEH 0	RCLK CDH 0	TCLK CCH 0	EXEN2 CBH 0	TR2 CAH 0	CNT2 C9H 0	CAP2 C8H 0	BITS	T2CON ¹ C8H 00H	RESERVED	RCAP2L CAH 00H	RCAP2H CBH 00H	TL2 CCH 00H	TH2 CDH 00H	RESERVED	RESERVED
PRE3 C7H 0	PRE2 C6H 0	PRE1 C5H 0	PRE0 C4H 1	WDIR C3H 0	WDS C2H 0	WDE C1H 0	WDWR C0H 0	BITS	WDCON ¹ C0H 10H	RESERVED	CHIPID C2H XXH	RESERVED	RESERVED	RESERVED	EDARL C6H 00H	EDARH C7H 00H
PSI BFH 0	PADC BEH 0	PT2 BDH 0	PS BCH 0	PT1 BBH 0	PX1 BAH 0	PT0 B9H 0	PX0 B8H 0	BITS	IP ¹ B8H 00H	ECON B9H 00H	RESERVED	RESERVED	EDATA1 BCH 00H	EDATA2 BDH 00H	EDATA3 BEH 00H	EDATA4 BFH 00H
RD B7H 1	WR B6H 1	T1 B5H 1	T0 B4H 1	INT1 B3H 1	INT0 B2H 1	TxD B1H 1	RxD B0H 1	BITS	P3 ¹ B0H FFH	PWM0L B1H 00H	PWM0H B2H 00H	PWM1L B3H 00H	PWM1H B4H 00H	NOT USED	NOT USED	SPH B7H 00H
EA AFH 0	EADC AEH 0	ET2 ADH 0	ES ACH 0	ET1 ABH 0	EX1 AAH 0	ET0 A9H 0	EX0 A8H 0	BITS	IE ¹ A8H 00H	IEIP2 A9H A0H	RESERVED	RESERVED	RESERVED	RESERVED	PWMCON AEH 00H	CFG841/ CFG842 AFH 00H
A7H 1	A6H 1	A5H 1	A4H 1	A3H 1	A2H 1	A1H 1	A0H 1	BITS	P2 ¹ A0H FFH	TIMECON A1H 00H	HTHSEC A2H 00H	SEC A3H 00H	MIN A4H 00H	HOURL A5H 00H	INTVAL A6H 00H	DPCON A7H 00H
SM0 9FH 0	SM1 9EH 0	SM2 9DH 0	REN 9CH 0	TB8 9BH 0	RB8 9AH 0	T1 99H 0	R1 98H 0	BITS	SCON ¹ 98H 00H	SBUF 99H 00H	I2CDAT 9AH 00H	I2CADD 9BH 55H	NOT USED	T3FD 9DH 00H	T3CON 9EH 00H	NOT USED
97H 1	96H 1	95H 1	94H 1	93H 1	92H 1	T2EX 91H 1	T2 90H 1	BITS	P1 ^{1,2} 90H FFH	I2CADD1 91H 7FH	I2CADD2 92H 7FH	I2CADD3 93H 7FH	NOT USED	NOT USED	NOT USED	NOT USED
TF1 8FH 0	TR1 8EH 0	TF0 8DH 0	TR0 8CH 0	IE1 8BH 0	IT1 8AH 0	IE0 89H 0	IT0 88H 0	BITS	TCON ¹ 88H 00H	TMOD 89H 00H	TL0 8AH 00H	TL1 8BH 00H	TH0 8CH 00H	TH1 8DH 00H	RESERVED	RESERVED
87H 1	86H 1	85H 1	84H 1	83H 1	82H 1	81H 1	80H 1	BITS	P0 ¹ 80H FFH	SP 81H 07H	DPL 82H 00H	DPH 83H 00H	DPP 84H 00H	RESERVED	RESERVED	PCON 87H 00H

Obr. B.10: Tabulka SFR ADuC841, ADuC842 a ADuC843 [13]

ISPI FFH 0	WCOL FEH 0	SPE FDH 0	SPIM FCH 0	CPOL FBH 0	CPHA FAH 1	SPR1 F9H 0	SPR0 F8H 0	BITS	SPICON F8H 05H	RESERVED	RESERVED	DACL FBH 00H	DACH FCH 00H	DACCON FDH 00H	RESERVED	RESERVED
F7H 0	F6H 0	F5H 0	F4H 0	F3H 0	F2H 0	F1H 0	F0H 0	BITS	B F0H 00H	RESERVED	I2CADD1 F2H 7FH	NOT USED	RESERVED	RESERVED	RESERVED	SPIDAT F7H 00H
MDO EFH 0	MDE EEH 0	MCO EDH 0	MDI ECH 0	I2CM EBH 0	I2CRS EAH 0	I2CTX E9H 0	I2CI E8H 0	BITS	I2CCON E8H 00H	GN0L ² E9H xxH	GN0M ² EAH xxH	GN0H ² EBH xxH	GN1L ² ADUC845 ONLY ECH xxH	GN1H ² ADUC845 ONLY EDH xxH	RESERVED	RESERVED
E7H 0	E6H 0	E5H 0	E4H 0	E3H 0	E2H 0	E1H 0	E0H 0	BITS	ACC E0H 00H	OF0L E1H xxH	OF0M E2H xxH	OF0H E3H xxH	OF1L ADUC845 ONLY E4H xxH	OF1H ADUC845 ONLY E5H xxH	ADC0CON2 E6H 00H	RESERVED
RDY0 DFH 0	RDY1 DEH 0	CAL DDH 0	NOXREF DCH 0	ERR0 DBH 0	ERR1 DAH 0	D9H 0	D8H 0	BITS	ADCSTAT D8H 00H	ADC0L NOT AVAILABLE ON ADUC848 D9H 00H	ADC0M DAH 00H	ADC0H DBH 00H	ADC1M ADUC845 ONLY DCH 00H	ADC1H ADUC845 ONLY DDH 00H	ADC1L ADUC845 ONLY DEH 00H	PSMCON DFH DEH
CY D7H 0	AC D6H 0	F0 D5H 0	RS1 D4H 0	RS0 D3H 0	OV D2H 0	FI D1H 0	P D0H 0	BITS	PSW D0H 00H	ADCMODE D1H 08H	ADC0CON1 D2H 07H	ADC1CON ADUC845 ONLY D3H 00H	SF D4H 45H	ICON D5H 00H	RESERVED	PLLCON D7H 53H
TF2 CFH 0	EXF2 CEH 0	RCLK CDH 0	TCLK CCH 0	EXEN2 CBH 0	TR2 CAH 0	CNT2 C9H 0	CAP2 C8H 0	BITS	T2CON C8H 00H	RESERVED	RCAP2L CAH 00H	RCAP2H CBH 00H	TL2 CCH 00H	TH2 CDH 00H	RESERVED	RESERVED
PRE3 C7H 0	PRE2 C6H 0	PRE1 C5H 0	PRE0 C4H 0	WDIR C3H 1	WDS C2H 0	WDE C1H 0	WDWR C0H 0	BITS	WDCON C0H 10H	RESERVED	CHIPID C2H A0H	RESERVED	RESERVED	RESERVED	EDARL C6H 00H	EDARH C7H 00H
BFH 0	PADC BEH 0	PT2 BDH 0	PS BCH 0	PT1 BBH 0	PX1 BAH 0	PT0 B9H 0	PX0 B8H 0	BITS	IP B8H 00H	ECON B9H 00H	RESERVED	RESERVED	EDATA1 BCH 00H	EDATA2 BDH 00H	EDATA3 BEH 00H	EDATA4 BFH 00H
RD B7H 1	WR B6H 1	T1 B5H 1	T0 B4H 1	INT1 B3H 1	INT0 B2H 1	TxD B1H 1	RxD B0H 1	BITS	P3 B0H FFH	PWM0L B1H 00H	PWM0H B2H 00H	PWM1L B3H 00H	PWM1H B4H 00H	RESERVED	RESERVED	SPH B7H 00H
EA AFH 0	EADC AEH 0	ET2 ADH 0	ES ACH 0	ET1 ABH 0	EX1 AAH 0	ET0 A9H 0	EX0 A8H 0	BITS	IE A8H 00H	IEIP2 A9H A 0H	RESERVED	RESERVED	RESERVED	RESERVED	PWMCON AEH 00H	CFG845/7/8 AFH 00H
A7H 1	A6H 1	A5H 1	A4H 1	A3H 1	A2H 1	A1H 1	A0H 1	BITS	P2 A0H FFH	TIMECON A1H 00H	HTHSEC ¹ A2H 00H	SEC ¹ A3H 00H	MIN ¹ A4H 00H	HOURL ¹ A5H 00H	INTVAL A6H 00H	DPCON A7H 00H
SM0 9FH 0	SM1 9EH 0	SM2 9DH 0	REN 9CH 0	TB8 9BH 0	RB8 9AH 0	T1 99H 0	RI 98H 0	BITS	SCON 98H 00H	SBUF 99H 00H	I2CDAT 9AH 00H	I2CADD 9BH 55H	RESERVED	T3FD 9DH 00H	T3CON 9EH 00H	EWAIT 9FH 00H
97H 1	96H 1	95H 1	94H 1	93H 1	92H 1	T2EX 91H 1	T2 90H 1	BITS	P1 90H FFH	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
TF1 8FH 0	TR1 8EH 0	TF0 8DH 0	TR0 8CH 0	IE1 8BH 0	IT1 8AH 0	IE0 89H 0	IT0 88H 0	BITS	TCON 88H 00H	TMOD 89H 00H	TL0 8AH 00H	TL1 8BH 00H	TH0 8CH 00H	TH1 8DH 00H	RESERVED	RESERVED
87H 1	86H 1	85H 1	84H 1	83H 1	82H 1	81H 1	80H 1	BITS	P0 80H FFH	SP 81H 07H	DPL 82H 00H	DPH 83H 00H	DPP 84H 00H	RESERVED	RESERVED	PCON 87H 00H

Obr. B.11: Tabulka SFR ADuC845, ADuC847 a ADuC848 [14]

Příloha C – Parametry SFR ADuC8xx

(mimo základní parametry 8051 a 8052) [5], [6], [7], [8], [9], [10], [11], [12], [13], [14]

C.1. ADC0CON (Primary ADC Control Reg. – řídicí slovo primárního A/D)

Mikrokontrolér: ADuC816, ADuC824, ADuC834, ADuC836

SFR adresa: D2h

Výchozí hodnota: 07h

Bitové adresace: Ne

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
-	XREF0	CH1	CH0	UNI0	RN2	RN1	RN0

XREF0 Primary ADC External Reference Select Bit – nastavení reference

1 – externí reference REF_{IN}

0 – interní reference $V_{REF}=1,25V$

CH0, CH1 Primary ADC Channel Selection Bits – nastavení diferenciálních párů

UNI0 Primary ADC Unipolar Bit – nastavení kódování

1 – unipolární kódování – nulový rozdíl vstupů = 0000h (000000h)

0 – bipolární kódování – nulový rozdíl vstupů = 8000h (800000h)

RN0 – RN2 Primary ADC Range Bits – rozsah primární A/D převodníku (při $V_{REF}=2,5V$)

CH1	CH0	kladný vstup	záporný vstup
0	0	AIN1	AIN2
0	1	AIN3	AIN4
1	0	AIN2	AIN2
1	1	AIN3	AIN2

RN2	RN1	RN0	rozsah
0	0	0	±20 mV
0	0	1	±40 mV
0	1	0	±80 mV
0	1	1	±160 mV
1	0	0	±320 mV
1	0	1	±640 mV
1	1	0	±1,28 V
1	1	1	±2,56 V

C.2. ADC0H/M/(L) (Primary ADC Conversion Result Reg. – data z prim. A/D)

Mikrokontrolér: ADuC816, ADuC836

SFR adresa: DBh/DAh

Výchozí hodnota: 00h/00h

Bitová adresace: Ne/Ne

V tomto 16bitovém slově je výsledek měření z primárního A/D převodníku.

Mikrokontrolér: ADuC824, ADuC834

SFR adresa: DBh/DAh/D9h

Výchozí hodnota: 00h/00h/00h

Bitová adresace: Ne/Ne/Ne

V tomto 24bitovém slově je výsledek měření z primárního A/D převodníku.

C.3. ADC1CON (Auxiliary ADC Control Reg. – řídicí slovo pomocného A/D)

Mikrokontrolér: ADuC816, ADuC824, ADuC834, ADuC836

SFR adresa: D3h

Výchozí hodnota: 00h

Bitové adresace: Ne

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
-	XREF1	ACH1	ACH0	UNI1	-	-	-

XREF1 Auxiliary ADC External Reference Select Bit – nastavení reference

1 – externí reference REF_{IN}

0 – interní reference $V_{REF}=1,25V$

ACH0, ACH1 Auxiliary ADC Channel Selection Bits – nastavení vstupů

ACH1	ACH0	kladný vstup	záporný vstup
0	0	AIN3	AGND
0	1	AIN4	AGND
1	0	Teplotní senzor	AGND
1	1	AIN5	AGND

UNI1 Auxiliary ADC Unipolar Bit – nastavení kódování

1 – unipolární kódování – nulový rozdíl vstupů = 0000h

0 – bipolární kódování – nulový rozdíl vstupů = 8000h

C.4. ADC1H/L (Auxiliary ADC Conversion Result Reg. – data z pomocn. A/D)

Mikrokontrolér: ADuC816, ADuC824, ADuC834, ADuC836

SFR adresa: DDh/DCh

Výchozí hodnota: 00h/00h

Bitová adresace: Ne/Ne

V tomto 16bitovém slově je výsledek měření z pomocného A/D převodníku.

C.5. ADCCON1 (ADC Control – řídicí slovo 1 A/D převodníku)

Mikrokontrolér: ADuC812

SFR adresa: EFh

Výchozí hodnota: 20h

Bitové adresace: Ne

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
MD1	MD0	CK1	CK0	AQ1	AQ0	T2C	EXC

MD0, MD1 Mode bits – volba provozu převodníku

CK0, CK1 ADC Clock Divide Bits – volba poměru dělení hlavních hodin pro hodiny A/D převodníku

AQ0, AQ1 ADC Acquisition Time Select Bits – nastavení času držení (HOLD) vstupním zesilovačem

MD1	MD0	
0	0	A/D převodník vypnut
0	1	A/D převodník v běžném provozu
1	0	A/D převodník vypnut, pokud neprobíhá měřicí cyklus
1	1	A/D převodník v pohotovostním režimu, pokud neprobíhá měřicí cyklus

CK1	CK0	dělitel MCLK
0	0	1
0	1	2
1	0	4
1	1	8

AQ1	AQ0	ADC Clks
0	0	1
0	1	2
1	0	4
1	1	8

T2C Timer 2 conversion bit – konverzní bit časovače 2 – pokud je nastaven tento bit, je při přetečení časovače 2, zahájen měřicí cyklus A/D převodníku

EXC External trigger enable bit – povolení externího spouštěcího vstupu \overline{EA} . Měřicí cyklus je zahájen při sestupné hraně (šířka pulzu > 100ns).

Mikrokontrolér: ADuC814, ADuC831, ADuC832, ADuC841, ADuC842, ADuC843

SFR adresa: EFh

Výchozí hodnota: 00h, 40h (ADuC841/2/3)

Bitové adresace: Ne

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
MD1	EXT_REF	CK1	CK0	AQ1	AQ0	T2C	EXC

MD1 Mode bit – volba provozu převodníku

0 – A/D převodník vypnut

1 – A/D převodník v běžném provozu

EXT_REF External Reference Select Bit – volba použití retenčního napětí pro A/D převodník

1 – externí reference

0 – interní reference

CK0, CK1 ADC Clock Divide Bits – volba poměru dělení PLL nebo základních hodin pro hodiny A/D převodníku (max 4,5 MHz)

AQ0, AQ1 ADC Acquisition Time Select Bits – nastavení času držení (HOLD) vstupním zesilovačem

CK1	CK0	ADuC814	ADuC831	ADuC832	ADuC841/2/3
0	0	8	16	8	32
0	1	4	2	4	4
1	0	16	4	16	8
1	1	32	8	32	2

AQ1	AQ0	ADC Clks
0	0	1
0	1	2
1	0	4
1	1	8

T2C Timer 2 conversion bit – konverzní bit časovače 2 – pokud je nastaven tento bit, je při přetečení časovače 2, zahájen měřicí cyklus A/D převodníku

EXC External trigger enable bit – povolení externího spouštěcího vstupu \overline{CONVST} . Měřicí cyklus je zahájen při sestupné hraně (šířka pulzu > 100ns).

C.6. ADCCON2 (ADC Control – řídicí slovo 2 A/D převodníku)

Mikrokontrolér: ADuC812, ADuC831, ADuC832, ADuC841, ADuC842, ADuC843

SFR adresa: D8h

Výchozí hodnota: 00h

Bitové adresace: Ano

bit 7 (DFh)	bit 6 (DEh)	bit 5 (DDh)	bit 4 (DCh)	bit 3 (DBh)	bit 2 (DAh)	bit 1 (D9h)	bit 0 (D8h)
ADCI	DMA	CCONV	SCONV	CS3	CS2	CS1	CS0

ADCI ADC interrupt bit – přerušení A/D převodníku – k nastavení do log. 1 dojde na konci jednoho cyklu měření A/D převodníku nebo na konci konverze bloku DMA. Zavoláním podprogramu přerušení dojde k nastavení do log. 0.

DMA DMA mode enable bit – uživatelsky nastavitelný bit pro uvolnění DMA režimu.

CCONV Continuous conversion bit – uživatelsky nastavitelný bit pro nepřetržitý provoz měřicího cyklu A/D převodníku. Měřicí cyklus A/D převodníku se spustí okamžitě po dokončení předchozího.

SCONV Single conversion bit – uživatelsky nastavitelný bit pro zahájení jednoho měřicího cyklu, po dokončení cyklu se tento bit nastaví do log. 0.

CS0 - CS3 Channel selection bits – bity výběru kanálu měření umožňující uživateli mít výběr softwarově podkontrolou. Při zahájení cyklu měření bude vybrán kanál podle volby těchto bitů. V režimu DMA je volba kanálů odvozená od ID zapsané v externí paměti.

CS3	CS2	CS1	CS0	Kanál
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	Senzor teploty
1	0	0	1	DAC0*
1	0	1	0	DAC1*
1	0	1	1	AGND*
1	1	0	0	V _{REF} *
1	1	1	1	DMA STOP

* tyto kanály nejsou použity u ADuC812

Mikrokontrolér: ADuC814

SFR adresa: D8h

Výchozí hodnota: 00h

Bitové adresace: Ano

bit 7 (DFh)	bit 6 (DEh)	bit 5 (DDh)	bit 4 (DCh)	bit 3 (DBh)	bit 2 (DAh)	bit 1 (D9h)	bit 0 (D8h)
ADCI	ADCSPI	CCONV	SCONV	CS3	CS2	CS1	CS0

ADCI ADC interrupt bit – přerušení A/D převodníku – k nastavení do log. 1 dojde na konci jednoho cyklu měření A/D převodníku nebo na konci konverze bloku DMA. Zavoláním podprogramu přerušení dojde k nastavení do log. 0.

ADCSPI ADCSPI Mode Enable Bit – při nastavení tohoto bitu výsledky měření jsou ukládány přímo do SPI paměti (SPIDAT) bez zásahu procesoru.

CCONV Continuous conversion bit – uživatelsky nastavitelný bit pro nepřetržitý provoz měřicího cyklu A/D převodníku. Měřicí cyklus A/D převodníku se spustí okamžitě po dokončení předchozího.

SCONV Single conversion bit – uživatelsky nastavitelný bit pro zahájení jednoho měřicího cyklu, po dokončení cyklu se tento bit nastaví do log. 0.

CS0 - CS3 Channel selection bits – bity výběru kanálu měření umožňující uživateli mít výběr softwarově podkontrolou. Při zahájení cyklu měření bude vybrán kanál podle volby těchto bitů. V režimu DMA je volba kanálů odvozená od ID zapsané v externí paměti.

CS3	CS2	CS1	CS0	Kanál
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	není
0	1	1	1	není
1	0	0	0	Senzor teploty
1	0	0	1	DAC0
1	0	1	0	DAC1
1	0	1	1	AGND
1	1	0	0	V _{REF}

C.7. ADCCON3 (ADC Control – řídicí slovo 3 A/D převodníku)

Mikrokontrolér: ADuC812, ADuC814, ADuC831, ADuC832, ADuC841, ADuC842, ADuC843

SFR adresa: F5h

Výchozí hodnota: 00h

Bitové adresace: Ne

ADuC812 – jen bit7 BUSY Ostatní bity jsou vyhrazeny pro vnitřní užití a budou vždy čteny jako log. 0 a při zápisu do těchto bitů, je také nutné nastavit log. 0.

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
BUSY	GNCLD	AVGS1	AVGS0	OFCLD	MODCAL	TYPECAL	SCAL

BUSY ADC busy status bit – bit určený jen pro čtení, který signalizuje průběh cyklu měření nebo kalibrace A/D převodníku. Příznak je po dokončení cyklu měření shozen do log. 0.

GNCLD Gain Calibration Disable Bit – tento bit zapíná (log. 0) / vypíná (log. 1) přepočtení hodnot A/D převodníku pomocí koeficientu zisku. (ADuC841, ADuC842 a ADuC843 vždy log. 0)

AVGS0, AVGS1 Number of Averages Selection Bits – výběr počtu měření A/D převodníkem během kalibračního měření

AVGS1	AVGS0	počet měření
0	0	15
0	1	1
1	0	31
1	1	63

OFCLD Offset Calibration Disable Bit – tento bit zapíná (log. 0) / vypíná (log. 1) přepočtení hodnot A/D převodníku pomocí kompenzačního koeficientu (nula). (ADuC831, ADuC832, ADuC841, ADuC842 a ADuC843 vždy log. 0)

MODCAL Calibration Mode Select Bit – tento bit musí být nastaven na log. 1 pro všechny kalibrační cykly.

TYPECAL Calibration Type Select Bit – volba typu kalibrace
0 – kalibrace kompenzačního koeficientu (nula stupnice)
1 – kalibrace koeficientu zisku (plný rozsah)

SCAL Start Calibration Cycle Bit – při nastavení tohoto bitu se spustí vybraný kalibrační cyklus. Po dokončení je automaticky vymazán.

C.8. ADCDATAH/L (ADC data registers – data z A/D převodníku)

Mikrokontrolér: ADuC812, ADuC814, ADuC831, ADuC832, ADuC841, ADuC842, ADuC843

SFR adresa: DAh/D9h

Výchozí hodnota: 00h/00h

Bitová adresace: Ne/Ne

ADCDATAH								ADCDATA L							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
CH-ID				výsledek měření A/D převodníku											

V tomto 16bitovém slově je výsledek měření A/D převodníku, kde horní 4 bity (CH-ID) je číslo změřeného vstupu a dolních 12 bitů je vlastní výsledek měření.

C.9. ADCGAINH/L (ADC Gain Calibratin Coefficient – zisk A/D převodníku)

Mikrokontrolér: ADuC812, ADuC814, ADuC831, ADuC832, ADuC841, ADuC842, ADuC843

SFR adresa: F4h/F3h

Výchozí hodnota: 00h/00h

Bitová adresace: Ne/Ne

ADCDATAH								ADCDATA L							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
-		koeficient zisku A/D převodníku													

V tomto 16bitovém slově je 14bitový koeficient zisku A/D převodníku, kde horní 2 bity jsou ignorovány a dolních 14 bitů je vlastní koeficient zisku.

C.10. ADCMODE (ADC Mode Register – režim A/D převodníku)

Mikrokontrolér: ADuC816, ADuC824, ADuC834, ADuC836

SFR adresa: D1h

Výchozí hodnota: 00h

Bitové adresace: Ne

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
-	-	ADC0EN	ADC1EN	-	MD2	MD1	MD0

ADC0EN Primary ADC Enable – zapnutí (log. 1) / vypnutí (log. 0) primárního A/D př.

ADC1EN Auxiliary ADC Enable – zapnutí (log. 1) / vypnutí (log. 0) pomocného A/D př.

MD0 – MD2 Primary and Auxiliary ADC Mode bits – volba provozu převodníků

MD2	MD1	MD0	
0	0	0	Vypnuto
0	0	1	Klidový režim
0	1	0	Jednotlivý měřicí režim
0	1	1	Kontinuální měřicí režim
1	0	0	Interní kalibrace nuly
1	0	1	Interní kalibrace zesílení
1	1	0	Systémová kalibrace nuly
1	1	1	Systémová kalibrace zesílení

C.11. ADCOFSH/L (ADC Offset Calibratin Coefficient – posun A/D převodníku)

Mikrokontrolér: ADuC812, ADuC814, ADuC831, ADuC832, ADuC841, ADuC842, ADuC843

SFR adresa: F2h/F1h

Výchozí hodnota: 00h/00h

Bitová adresace: Ne/Ne

ADCDATAH						ADCDATAL									
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
-						-						koeficient posunu A/D převodníku („nula“)			

V tomto 16bitovém slově je 14bitový koeficient posunu A/D převodníku, kde horní 2 bity jsou ignorovány a dolních 14 bitů je vlastní koeficient posunu.

C.12. ADCSTAT (ADC Status Register – stavový registr)

Mikrokontrolér: ADuC816, ADuC824, ADuC834, ADuC836

SFR adresa: D8h

Výchozí hodnota: 00h

Bitové adresace: Ano

bit 7 (DFh)	bit 6 (DEh)	bit 5 (DDh)	bit 4 (DCh)	bit 3 (DBh)	bit 2 (DAh)	bit 1 (D9h)	bit 0 (D8h)
RDY0	RDY1	CAL	NOXREF	ERR0	ERR1	-	-

RDY0 Ready Bit for Primary ADC – tento bit je nastaven do log. 1 po dokončení měřicího nebo kalibračního cyklu primárního A/D převodníku. Vymazání se provádí příkazem nebo automaticky při startu nového měřicího nebo kalibračního cyklu.

RDY1 Ready Bit for Auxiliary ADC – stejná definice jako RDY0, jen pro pomocný A/D převodník.

CAL Calibration Status Bit – tento bit je nastaven do log. 1 po dokončení kalibračního cyklu A/D převodníku. Vymazání se provádí příkazem nebo automaticky při startu nového měřicího nebo kalibračního cyklu.

NOXREF No External Reference Bit (aktivní pouze v případě, že primární nebo pomocný A/D převodník je aktivní) – pokud je tento bit v log. 1 je jeden nebo oba REFIN vstupy mimo povolenou mez. Log. 0 = V_{REF} je v pořádku.

ERR0 Primary ADC Error Bit – pokud je bit v log. 1, může být výsledek měření mimo rozsah (ADC0=0h nebo FFFFh) nebo po kalibraci nebylo možno uložit nové konstanty. Vynuluje se při zahájení nového cyklu měření nebo kalibrace.

ERR1 Auxiliary ADC Error Bit ADC – stejná definice jako ERR0, jen pro pomocný A/D převodník.

C.13. CFG814 (ADuC814 Configuration Register – konfigurace)

Mikrokontrolér: ADuC814

SFR adresa: 9Ch

Výchozí hodnota: 04h

Bitové adresace: Ne

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
-	-	-	-	-	-	EXTCLK	SER_EN

EXTCLK External Clock Selection Bit – volba základních hodin

0 – základní hodiny XTAL a PLL

1 – základní hodiny EXTCLK

SER_EN Serial Interface Enable Bit – povolení sériového rozhraní

0 – standardní funkce vývodů P3.5, P3.6 a P3.7

1 – SPI rozhraní na vývodech P3.5, P3.6 a P3.7

C.14. CFG831 (ADuC831 Configuration Register – konfigurace)

Mikrokontrolér: ADuC831

SFR adresa: AFh

Výchozí hodnota: 10h (pro krystal s taktovací frekvencí 11,0592 MHz)

Bitové adresace: Ne

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
EXSP	PWPO	DBUF	EPM2	EPM1	EPM0	-	XRAMEN

EXSP Extended SP Enable – povolení použití SPH

PWPO PWM Pin Out Selection – volba PWM výstupů

0 – pro PWM jsou použity výstupy P2.6 a P2.7

1 – pro PWM jsou použity výstupy P3.3 a P3.4

DBUF DAC Output Buffer – volba použití výstupního zásobníku D/A převodníku

0 – výstupní zásobník je použit

1 – výstupní zásobník je vynechán

EPM0 – EPM2 Flash/EE Controller and PWM Clock Frequency Configuration Bits – nastavení frekvence pro Flash/EE a PWM. Frekvence by se po vydělení koeficientem měla rovnat $32 \text{ kHz} \pm 50 \%$

EPM2	EPM1	EPM0	koeficient
0	0	0	32
0	0	1	64
0	1	0	128
0	1	1	256
1	0	0	512
1	0	1	1024

XRAMEN XRAM Enable Bit – volba použití XRAM

1 – interní XRAM je namapována do dolních 2 kB externí paměti

0 – interní XRAM není přístupná

C.15. CFG832 (ADuC832 Configuration Register – konfigurace)**Mikrokontrolér: ADuC832**

SFR adresa: AFh

Výchozí hodnota: 00h

Bitové adresace: Ne

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
EXSP	PWPO	DBUF	EXTCLK	-	-	-	XRAMEN

EXSP Extended SP Enable – povolení použít SPH

PWPO PWM Pin Out Selection – volba PWM výstupů

0 – pro PWM jsou použity výstupy P2.6 a P2.7

1 – pro PWM jsou použity výstupy P3.3 a P3.4

DBUF DAC Output Buffer – volba použití výstupního zásobníku D/A převodníku

0 – výstupní zásobník je použit

1 – výstupní zásobník je vynechán

EXTCLK volba použití externích hodin

1 – volba použití externích hodin (P3.4)

0 – volba použití interních hodin PLL

XRAMEN XRAM Enable Bit – volba použití XRAM

1 – interní XRAM je namapována do dolních 2 kB externí paměti

0 – interní XRAM není přístupná

C.16. CFG834 (ADuC834 Configuration Register – konfigurace)**Mikrokontrolér: ADuC834**

SFR adresa: AFh

Výchozí hodnota: 00h

Bitové adresace: Ne

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
EXSP	-	-	-	-	-	-	XRAMEN

EXSP Extended SP Enable – povolení použít SPH

XRAMEN XRAM Enable Bit – volba použití XRAM

1 – interní XRAM je namapována do dolních 2kB externí paměti

0 – interní XRAM není přístupná

C.17. CFG836 (ADuC836 Configuration Register – konfigurace)**Mikrokontrolér: ADuC836**

SFR adresa: AFh

Výchozí hodnota: 00h

Bitové adresace: Ne

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
EXSP	-	-	-	-	-	-	XRAMEN

EXSP Extended SP Enable – povolení použít SPH

XRAMEN XRAM Enable Bit – volba použití XRAM

1 – interní XRAM je namapována do dolních 2 kB externí paměti

0 – interní XRAM není přístupná

C.18. CFG841 (ADuC841 Configuration Register – konfigurace)

Mikrokontrolér: ADuC841

SFR adresa: AFh

Výchozí hodnota: 10h (pro krystal s taktovací frekvencí 11,0592 MHz)

Bitové adresace: Ne

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
EXSP	PWPO	DBUF	EPM2	EPM1	EPM0	-	XRAMEN

EXSP Extended SP Enable – povolení použít SPH

PWPO PWM Pin Out Selection – volba PWM výstupů

0 – pro PWM jsou použity výstupy P2.6 a P2.7

1 – pro PWM jsou použity výstupy P3.3 a P3.4

DBUF DAC Output Buffer – volba použití výstupního zásobníku D/A převodníku

0 – výstupní zásobník je použit

1 – výstupní zásobník je vynechán

EPM0 – EPM2 Flash/EE Controller and PWM Clock Frequency Configuration Bits – nastavení frekvence pro Flash/EE a PWM. Frekvence by se po vydělení koeficientem měla rovnat $32 \text{ kHz} \pm 50 \%$

EPM2	EPM1	EPM0	koeficient
0	0	0	32
0	0	1	64
0	1	0	128
0	1	1	256
1	0	0	512
1	0	1	1024

XRAMEN XRAM Enable Bit – volba použití XRAM

1 – interní XRAM je namapována do dolních 2 kB externí paměti

0 – interní XRAM není přístupná

C.19. CFG842 (ADuC842 Configuration Register – konfigurace)

Mikrokontrolér: ADuC842, ADuC843

SFR adresa: AFh

Výchozí hodnota: 00h

Bitové adresace: Ne

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
EXSP	PWPO	DBUF	EXTCLK	-	-	-	XRAMEN

EXSP Extended SP Enable – povolení použít SPH

PWPO PWM Pin Out Selection – volba PWM výstupů

0 – pro PWM jsou použity výstupy P2.6 a P2.7

1 – pro PWM jsou použity výstupy P3.3 a P3.4

DBUF DAC Output Buffer – volba použití výstupního zásobníku D/A převodníku

0 – výstupní zásobník je použit

1 – výstupní zásobník je vynechán

EXTCLK volba použití externích hodin

1 – volba použití externích hodin (P3.4)

0 – volba použití interních hodin PLL

XRAMEN XRAM Enable Bit – volba použití XRAM

1 – interní XRAM je namapována do dolních 2 kB externí paměti

0 – interní XRAM není přístupná

C.20. DACCON (DAC Control – řídicí slovo D/A převodníku)

Mikrokontrolér: ADuC812, ADuC814, ADuC831, ADuC832, ADuC841, ADuC842

SFR adresa: FDh

Výchozí hodnota: 04h

Bitové adresace: Ne

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
MODE	RNG1	RNG0	/CLR1	/CLR0	SYNC	/PD1	/PD0

MODE	DAC Mode bit – nastaví režimu všech D/A převodníků 0 – 12bitový režim 1 – 8bitový režim (DACxH je ignorováno)
RNG0, RNG1	Range Select Bit DAC0/1 – výběr rozsahu D/A převodníku 0/1 0 – rozsah 0 – V_{REF} 1 – rozsah 0 – V_{DD}
/CLR0, /CLR1	Clear Bit DAC0/1 – „vymazání“ D/A převodníku 0/1 0 – výstup D/A převodníku nuceně na 0 V 1 – výstup D/A převodníku v normálním režimu
SYNC	Update Synchronization Bit – pokud je tento bit v log. 1 je stav DACxH/L přepisován okamžitě na výstup z D/A převodníku. V log.0 je možné DACxH/L přepisovat a výstup se nemění dokud není SYNC opět nastaven na log. 1.
/PD0, /PD1	Power-Down Bit DAC0/1 – ovládání napájení D/A převodníku 0/1 1 – napájení zapnuto 0 – napájení vypnuto

Mikrokontrolér: ADuC816, ADuC824, ADuC834, ADuC836

SFR adresa: FDh

Výchozí hodnota: 00h

Bitové adresace: Ne

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
-	-	-	DACPIN	DAC8	DACRN	/DACCLR	DACEN

DACPIN	DAC Output Pin Select – výběr použitého vývodu pro D/A převodník 1 – vývod 12 (P1.7/AIN4/DAC) 0 – vývod 3 (P1.2/DAC/IEXC1)
DAC8	DAC 8-bit Mode bit – nastaví režim všech D/A převodníků 0 – 12bitový režim 1 – 8bitový režim (DACH je ignorováno)
DACRN	DAC Output Range Bit – výběr rozsahu D/A převodníku 1 0 – rozsah 0 – V_{REF} 1 – rozsah 0 – V_{DD}
/DACCLR	DAC Clear Bit – „vymazání“ D/A převodníku 1 0 – výstup D/A převodníku nuceně na 0 V 1 – výstup D/A převodníku v normálním režimu
DACEN	DAC Enable Bit – ovládání napájení D/A převodníku 1 1 – napájení zapnuto 0 – napájení vypnuto

C.21. DACH/L (DAC Data Reg. – data do D/A převodníku)

Mikrokontrolér: ADuC816, ADuC824, ADuC834, ADuC836

SFR adresa: FBh (DACL), FCh (DACH)

Výchozí hodnota: 00h/00h

Bitové adresace: Ne/Ne

Při 8bitovém převodu (DAC8=1) se využívají jen DACL.

Při 12bitovém převodu (DAC8=0) je využíváno z DACH dolní 4 bity.

DACH								DACL							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
-								data do D/A převodníku							

C.22. DACxH/L (DAC Data Reg. – data do D/A převodníku)

Mikrokontrolér: ADuC812, ADuC814, ADuC831, ADuC832, ADuC841, ADuC842

SFR adresa: F9h (DAC0L), FAh (DAC0H), FBh (DAC1L), FCh (DAC1H)

Výchozí hodnota: 00h (všechny registry)

Bitové adresace: Ne (všechny registry)

Při 8bitovém převodu (MODE = 1) se využívají jen DACxL. Při 12bitovém převodu (MODE = 0) je využíváno z DACxH dolní 4bity a horní 4bity jsou ignorovány.

DACxH								DACxL							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
								data do D/A převodníku							

C.23. DMAL/H/P (DMA Address pointer – ukazatel DMA tabulky)

Mikrokontrolér: ADuC812, ADuC831, ADuC832, ADuC841, ADuC842, ADuC843

SFR adresa: D2h (DMAL), D3h (DMAH), D4h (DMAP)

Výchozí hodnota: 00h (všechny registry)

Bitové adresace: Ne (všechny registry)

Ukazatel DMA tabulky do které se vepisují výsledky měření v DMA režimu. Je třeba zachovat pořadí zápisu do registrů (DMAL → DMAH → DMAP).

C.24. ECON (Flash/EE data space Controls – řídicí slovo)

Mikrokontrolér: ADuC8xx

SFR adresa: B9h

Výchozí hodnota: 00h

Bitové adresace: Ne

01h READ COMMAND – Příkaz číst – načte se obsah stránky Flash/EE do EDATA1 – 4, kde adresa stránky je v EDARL/(H).

02h PROGRAM (WRITE) COMMAND – Příkaz programovat (zapisovat) – uloží se obsah EDATA1 – 4 do stránky Flash/EE adresovanou EDARL/(H). Předpokládá se, že stránka je vymazána.

03h RESERVED FOR INTERNAL USE – Vyhrazeno pro vnitřní použití – programově nesmí být použit.

04h VERIFY COMMAND – Příkaz ověření – umožňuje ověřit, zda údaje v EDATA1 – 4 odpovídá obsahu stránky Flash/EE adresovanou EDARL/(H). Pokud se obsah shoduje je při následném čtení ECON = 0, v opačném případě je ECON <> 0.

05h ERASE COMMAND – Příkaz vymazání – dojde k vymazání 4bytové stránky adresované EDARL.

06h ERASE ALL COMMAND – Příkaz vymazání všeho – dojde k vymazání všech 160 stránek Flash/EE (640 byte) nebo všech 1024 stránek Flash/EE (4096 byte) u ADuC83x a ADuC84x.

Následující příkazy jsou implementovány jen v ADuC83x a ADuC84x.

81h READBYTE COMMAND – Příkaz číst byte – načte se obsah byte Flash/EE do EDATA1, kde adresa byte je v EDARL/H.

82h WRITEBYTE COMMAND – Příkaz zapisovat byte – uloží se obsah EDATA1 do byte Flash/EE adresovanou EDARL/H.

C.25. EDARL/(H) (Flash/EE data page address – stránka Flash/EE)

Mikrokontrolér: ADuC81x, ADuC824

SFR adresa: C6h

Výchozí hodnota: 00h

Bitové adresace: Ne

Adresa 4bytové stránky Flash/EE. Povolený obsah je 00h – 9Fh (160 stránek = 640 byte)

Mikrokontrolér: ADuC83x, ADuC84x

SFR adresa: C6h/C7h

Výchozí hodnota: 00h/00h

Bitové adresace: Ne/Ne

Adresa 4bytové stránky Flash/EE. Povolený obsah je 0000h – 03FFh (1024 stránek = 4096 byte)

Nebo v bytovém režimu adresa v Flash/EE, kde povolený obsah je 0000h – 0FFFh.

C.26. EDATA1 – 4 (Flash/EE data)

Mikrokontrolér: ADuC8xx

SFR adresa: BCh (EDATA1), BDh (EDATA2), BEh (EDATA3), BFh (EDATA4)

Výchozí hodnota: 00h (všechny registry)

Bitové adresace: Ne (všechny registry)

Pomocné datové proměnné pro zápis a čtení Flash/EE.

C.27. ETIM1 – 3 (Flash/EE Memory Timig)

Mikrokontrolér: ADuC812, ADuC814

SFR adresa: BAh (ETIM1), BBh (ETIM2), C4h (ETIM3)

Výchozí hodnota: ETIM1=52h, ETIM2=04h, ETIM3=C9h

Bitové adresace: Ne (všechny registry)

Tyto registry ovlivňují rychlost zápisu/mazání pro Flash/EE.

Nastavení těchto registrů by mělo odpovídat následujícímu vztahu:

$ETIM2, ETIM1 = 100 \mu s \times f_{CLK}$ (ETIM2 = high byte, ETIM1 = low byte)

Při použití taktovací frekvence 11,0592 Mhz nemusíme ETIM1,2 měnit (odpovídá výchozí hodnotě)

Hodnota ETIM3 by měla zůstat vždy výchozí (C9h) (tento parametr jen u ADuC812).

C.28. GN0H/M/(L) (Primary ADC Gain Calibration – zisk primárního A/D)

Mikrokontrolér: ADuC816, ADuC836

SFR adresa: EBh/EAh

Výchozí hodnota: Podle předchozí kalibrace – první kalibrace provedena při výrobě

Bitová adresace: Ne/Ne

V tomto 16bitovém slově je koeficient zisku A/D převodníku.

Mikrokontrolér: ADuC824, ADuC834

SFR adresa: EBh/EAh/E9h

Výchozí hodnota: Podle předchozí kalibrace – první kalibrace provedena při výrobě

Bitová adresace: Ne/Ne/Ne

V tomto 24bitovém slově je koeficient zisku A/D převodníku.

C.29. GN1H/L (Auxiliary ADC Gain Calibration – zisk pomocného A/D)

Mikrokontrolér: ADuC816, ADuC824, ADuC834, ADuC836

SFR adresa: EDh/ECh

Výchozí hodnota: Podle předchozí kalibrace – první kalibrace provedena při výrobě

Bitová adresace: Ne/Ne

V tomto 16bitovém slově je koeficient zisku A/D převodníku.

C.30. HOUR (Hours Time Register – registr hodin)

Mikrokontrolér: ADuC814, ADuC816, ADuC824, ADuC83x, ADuC84x

SFR adresa: A5h

Výchozí hodnota: 00h

Bitová adresace: Ne

Hodnota v tomto registru se zvýší jednou za hodinu při TCEN = 1.

Pokud je TFH = 1 čítá HOUR od 0 do 23. Pokud je TFH = 0 čítá HOUR od 0 do 255.

C.31. HTHSEC (Hundredths Seconds Time Register – registr setin vteřiny)

Mikrokontrolér: ADuC814, ADuC816, ADuC824, ADuC83x, ADuC84x

SFR adresa: A2h

Výchozí hodnota: 00h

Bitová adresace: Ne

Hodnota v tomto registru se zvýší jednou za 1/128 sekundy při TCEN = 1. HTHSEC čítá od 0 do 127.

C.32. I2CADD (I²C Address Register – I²C adresa)

Mikrokontrolér: ADuC8xx

SFR adresa: 9Bh

Výchozí hodnota: 55h

Bitové adresace: Ne

Identifikující (první) adresa pro I²C přenos.

C.33. I2CADD1 (I²C Address Register – I²C adresa)

Mikrokontrolér: ADuC84x

SFR adresa: 91h

Výchozí hodnota: 7Fh

Bitové adresace: Ne

Identifikující druhá adresa pro I²C přenos.

C.34. I2CADD2 (I²C Address Register – I²C adresa)

Mikrokontrolér: ADuC841, ADuC842, ADuC843

SFR adresa: 92h

Výchozí hodnota: 7Fh

Bitové adresace: Ne

Identifikující třetí adresa pro I²C přenos.

C.35. I2CADD3 (I²C Address Register – I²C adresa)

Mikrokontrolér: ADuC841, ADuC842, ADuC843

SFR adresa: 93h

Výchozí hodnota: 7Fh

Bitové adresace: Ne

Identifikující čtvrtá adresa pro I²C přenos.

C.36. I2CCON (I²C Control Register – řídicí slovo I²C)**Mikrokontrolér: ADuC81x, ADuC824, ADuC83x, ADuC845, ADuC847, ADuC848**

SFR adresa: E8h

Výchozí hodnota: 00h

Bitové adresace: Ano

bit 7 (EFh)	bit 6 (EEh)	bit 5 (EDh)	bit 4 (ECh)	bit 3 (EBh)	bit 2 (EAh)	bit 1 (E9h)	bit 0 (E8h)
MDO	MDE	MCO	MDI	I2CM	I2CRS	I2CTX	I2CI

MDO I²C Software Master Data Output Bit (Master Mode Only) – tento bit se používá k implementaci I²C výstupu v software. Obsah tohoto bitu se přepíše na výstup SDATA, pokud je MDE = 1.

MDE I²C Software Master Data Output Enable Bit (Master Mode Only) – tento bit určuje směr přenosu:

1 – vývod SDATA je použit jako výstup (Tx)

0 – vývod SDATA je použit jako vstup (Rx)

MCO I²C Software Master Data Output Bit (Master Mode Only) – tento bit se používá k implementaci I²C výstupu v software. Obsah tohoto bitu se přepíše na vývod SCLOCK.

MDI I²C Software Master Data Input Bit (Master Mode Only) - tento bit se používá k implementaci I²C vstupu v software. Vstup SDATA se přepíše do tohoto bitu, pokud je MDE = 0.

I2CM I²C Master/Slave Mode Bit – nastavení režimu I²C

0 – slave mode – komunikace řízená hardware

1 – master mode – komunikace řízená software

I2CRS I²C Reset Bit (Slave Mode Only)

1 – reset I²C rozhraní0 – normální režim I²C

I2CTX I²C Direction Transfer Bit (Slave Mode Only) – tento bit určuje směr přenosu:

1 – rozhraní I²C vysílá data0 – rozhraní I²C přijímá data

I2CI I²C Interrupt Bit (Slave Mode Only) – tento bit je nastaven do log. 1 po dokončení odeslání nebo příjmu byte. Nastavit tento bit do log. 0 je nutno softwarově.

Mikrokontrolér: ADuC841, ADuC842, ADuC843

SFR adresa: E8h

Výchozí hodnota: 00h

Bitové adresace: Ano

Master mode:

bit 7 (EFh)	bit 6 (EEh)	bit 5 (EDh)	bit 4 (ECh)	bit 3 (EBh)	bit 2 (EAh)	bit 1 (E9h)	bit 0 (E8h)
MDO	MDE	MCO	MDI	I2CM	-	-	-

MDO I²C Software Master Data Output Bit (Master Mode Only) – tento bit se používá k implementaci I²C výstupu v software. Obsah tohoto bitu se přepíše na výstup SDATA, pokud je MDE = 1.

MDE I²C Software Master Data Output Enable Bit (Master Mode Only) – tento bit určuje směr přenosu:

1 – vývod SDATA je použit jako výstup (Tx)

0 – vývod SDATA je použit jako vstup (Rx)

MCO I²C Software Master Data Output Bit (Master Mode Only) – tento bit se používá k implementaci I²C výstupu v software. Obsah tohoto bitu se přepíše na vývod SCLOCK.

MDI I²C Software Master Data Input Bit (Master Mode Only) – tento bit se používá k implementaci I²C vstupu v software. Vstup SDATA se přepíše do tohoto bitu, pokud je MDE = 0.

I2CM I²C Master/Slave Mode Bit – nastavení režimu I²C

0 – slave mode – komunikace řízená hardware

1 – master mode – komunikace řízená software

Slave mode:

bit 7 (EFh)	bit 6 (EEh)	bit 5 (EDh)	bit 4 (ECh)	bit 3 (EBh)	bit 2 (EAh)	bit 1 (E9h)	bit 0 (E8h)
I2CSI	I2CGC	I2CID1	I2CID0	I2CM	I2CRS	I2CTX	I2CI

I2CSI I2C Stop Interrupt Enable Bit – pokud je tento bit nastaven a přijde stop bit I2C je vyvoláno přerušení I2C.

I2CGC I2C General Call Status Bit – tento bit je nastaven po příchodu volání s generální adresou. Vymazat lze jen příkazem.

I2CID0, I2CID1 I2C Interrupt Decode Bits – volba přerušení I2C

I2CID1	I2CID0	
0	0	start a odpovídající adresa
0	1	opakovaný start a odpovídající adresa
1	0	uživatelská data
1	1	stop po spuštění a odpovídající adresa

I2CM I²C Master/Slave Mode Bit – nastavení režimu I²C

0 – slave mode – komunikace řízená hardware

1 – master mode – komunikace řízená software

I2CRS I²C Reset Bit (Slave Mode Only)

1 – reset I²C rozhraní

0 – normální režim I²C

I2CTX I²C Direction Transfer Bit (Slave Mode Only) – tento bit určuje směr přenosu:

1 – rozhraní I²C vysílá data

0 – rozhraní I²C přijímá data

I2CI I²C Interrupt Bit (Slave Mode Only) – tento bit je nastaven do log. 1 po dokončení odeslání nebo příjmu byte. Nastavit tento bit do log. 0 je nutno softwarově.

C.37. I2CDAT (I²C Data Register – data pro I²C přenos)

Mikrokontrolér: ADuC8xx

SFR adresa: 9Ah

Výchozí hodnota: 00h

Bitové adresace: Ne

Vlastní data přenášená pomocí I²C rozhraní. K těmto datům by se mělo přistupovat jen jednou za cyklus přerušení.

C.38. IE (Interrupt Enable – povolení přerušení)

Mikrokontrolér: ADuC8xx

SFR adresa: A8h

Výchozí hodnota: 00h

Bitové adresace: Ano

bit 7 (AFh)	bit 6 (AEh)	bit 5 (ADh)	bit 4 (ACh)	bit 3 (ABh)	bit 2 (AAh)	bit 1 (A9h)	bit 0 (A8h)
EA	EADC	ET2	ES	ET1	EX1	ET0	EX0

EA Povolení (log. 1) / zákaz (log. 0) všech přerušení

EADC Povolení (log. 1) / zákaz (log. 0) přerušení A/D převodníků

ET2 Povolení (log. 1) / zákaz (log. 0) přerušení časovače 2

ES Povolení (log. 1) / zákaz (log. 0) přerušení sériového portu (UART)

ET1 Povolení (log. 1) / zákaz (log. 0) přerušení časovače 1

EX1 Povolení (log. 1) / zákaz (log. 0) vnějšího přerušení 1

ET0 Povolení (log. 1) / zákaz (log. 0) přerušení časovače 0

EX0 Povolení (log. 1) / zákaz (log. 0) vnějšího přerušení 0

C.39. IE2 (Secondary Interrupt Enable – povolení přerušení 2)**Mikrokontrolér:** ADuC812

SFR adresa: A9h

Výchozí hodnota: 00h

Bitové adresace: Ne

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
-	-	-	-	-	-	EPSMI	ESI

EPSMI Povolení (log. 1) / zákaz (log. 0) přerušení monitoru napájení

ESI Povolení (log. 1) / zákaz (log. 0) přerušení I²C/SPI**C.40. IEIP2 (Secondary Interrupt Enable and Priority – povolení a priorita přerušení 2)****Mikrokontrolér:** ADuC814, ADuC816, ADuC824, ADuC83x, ADuC84x

SFR adresa: A9h

Výchozí hodnota: A0h

Bitové adresace: Ne

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
-	PTI	PPSMI	PSI	0	ETI	EPSMI	ESI

PTI Vyšší (log. 1) / nižší (log. 0) priorita přerušení TIC

PPSMI Vyšší (log. 1) / nižší (log. 0) priorita přerušení monitoru napájení

PSI Vyšší (log. 1) / nižší (log. 0) priorita přerušení I²C/SPI

ETI Povolení (log. 1) / zákaz (log. 0) přerušení TIC

EPSMI Povolení (log. 1) / zákaz (log. 0) přerušení monitoru napájení

ESI Povolení (log. 1) / zákaz (log. 0) přerušení I²C/SPI**C.41. INTVAL (User Time Interval Select – uživatelsky nastavitelný časovač)****Mikrokontrolér:** ADuC814, ADuC816, ADuC824, ADuC83x, ADuC84x

SFR adresa: A6h

Výchozí hodnota: 00h

Bitová adresace: Ne

Hodnota v tomto registru je předvolba pro vnitřní čítač, který čítá na základě předvoleb v TIMECON.

Pokud se hodnota vnitřního čítače rovná INTVAL dojde k přerušení (viz C.40).

C.42. IP (Interrupt Priority – priorita přerušení)**Mikrokontrolér:** ADuC8xx

SFR adresa: B8h

Výchozí hodnota: 00h

Bitové adresace: Ano

bit 7 (BFh)	bit 6 (BEh)	bit 5 (BDh)	bit 4 (BCh)	bit 3 (BBh)	bit 2 (BAh)	bit 1 (B9h)	bit 0 (B8h)
PSI	PADC	PT2	PS	PT1	PX1	PT0	PX0

PSI (jen ADuC812) Vyšší (log. 1) / nižší (log. 0) priorita přerušení I²C/SPI

PADC Vyšší (log. 1) / nižší (log. 0) priorita přerušení A/D převodníků

PT2 Vyšší (log. 1) / nižší (log. 0) priorita přerušení časovače 2

PS Vyšší (log. 1) / nižší (log. 0) priorita přerušení sériového portu (UART)

PT1 Vyšší (log. 1) / nižší (log. 0) priorita přerušení časovače 1

PX1 Vyšší (log. 1) / nižší (log. 0) priorita vnějšího přerušení 1

PT0 Vyšší (log. 1) / nižší (log. 0) priorita přerušení časovače 0

PX0 Vyšší (log. 1) / nižší (log. 0) priorita vnějšího přerušení 0

C.43. MIN (Minutes Time Register – registr minut)**Mikrokontrolér:** ADuC814, ADuC816, ADuC824, ADuC83x, ADuC84x

SFR adresa: A4h

Výchozí hodnota: 00h

Bitová adresace: Ne

Hodnota v tomto registru se zvýší jednou za minutu při TCEN = 1. MIN čítá od 0 do 59.

C.44. OF0H/M/(L) (Primary ADC Offset Calibration – posun primárního A/D)**Mikrokontrolér:** ADuC816, ADuC836

SFR adresa: E3h/E2h

Výchozí hodnota: 80h/00h

Bitová adresace: Ne/Ne

V tomto 16bitovém slově je koeficient posunu A/D převodníku.

Mikrokontrolér: ADuC824, ADuC834

SFR adresa: E3h/E2h/E1h

Výchozí hodnota: 80h/00h/00h

Bitová adresace: Ne/Ne/Ne

V tomto 24bitovém slově je koeficient posunu A/D převodníku.

C.45. OF1H/L (Auxiliary ADC Offset Calibration – posun pomocného A/D)**Mikrokontrolér:** ADuC816, ADuC824, ADuC834, ADuC836

SFR adresa: E5h/E4h

Výchozí hodnota: 80h/00h

Bitová adresace: Ne/Ne

V tomto 16bitovém slově je koeficient posunu A/D převodníku.

C.46. PLLCON (PLL Control Register – řídicí slovo PLL)**Mikrokontrolér:** ADuC814, ADuC816, ADuC824, ADuC832, ADuC834, ADuC836, ADuC84x

SFR adresa: D7h

Výchozí hodnota: 03h

Bitové adresace: Ne

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
OSC_DP	LOCK	-	/LTEA	FINT	CD2	CD1	CD0

OSC_PD Oscillator Power-Down Bit – funkce oscilátoru v Power-Down režimu

1 – oscilátor se zastaví

0 – oscilátor běží dál (pokračuje stopování v TIC)

LOCK PLL

Lock Bit – jen pro čtení

1 – PLL je správně synchronizován s externím krystalem

0 – PLL není správně synchronizován s externím krystalem, frekvence PLL výstupu je 16,78 MHz ± 20 %

FINT

Fast Interrupt Response Bit – pokud je nastaven na log. 1 provádí se podprogram přerušení na nejrychlejší frekvenci jádra, bez ohledu na nastavení CD0 – CD2. Po návratu z podprogramu přerušení pokračuje provádění programu na přednastavené frekvenci jádra.

/LTEA

V tomto bitu je stav EĀ při resetu nebo zapnutí. (není v ADuC814)

CD0 – CD2

(Core Clock) Divider Bits – volba frekvence jádra [MHz]

CD2	CD1	CD0	ADuC814	ADuC8xx
0	0	0	16,777216	12,582912
0	0	1	8,388608	6,291456
0	1	0	4,194304	3,145728
0	1	1	2,097152	1,572864
1	0	0	1,048576	0,786432
1	0	1	0,524288	0,393216
1	1	0	0,262144	0,196608
1	1	1	0,131072	0,098304

C.47. PSMCON (Power Supply Monitor – monitor napájení)**Mikrokontrolér:** ADuC812, ADuC814, ADuC831, ADuC832, ADuC84x

SFR adresa: DFh

Výchozí hodnota: DCh

Bitové adresace: Ne

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
-	CMP	PSMI	TP2	TP1	TP0	PSF	PSMEN

CMP Comparator bit – tento bit je jen pro čtení a přímo odráží stav komparátorů napájení AV_{DD} a DV_{DD} .

1 – obě napětí (AV_{DD} a DV_{DD}) jsou nad nastavenou mez0 – jedno z napětí (AV_{DD} nebo DV_{DD}) je pod nastavenou mez

PSMI Power Supply Monitor Interrupt Bit – tento bit je nastaven log. 1 pokud je $CMP = 0$, v případě $CMP = 1$ je po 256 ms tento bit nastaven na log. 0 nebo je možné nastavit log. 0 uživatelem.

TP0 - TP2 Trip Point Selection Bits – výběr rozhodovací úrovně (u ADuC814 není TP2):

TP2	TP1	TP0	Rozhodovací úroveň [V]
0	0	0	4,63
0	0	1	4,37
0	1	0	3,08
0	1	1	2,93
1	0	0	2,63

PSF AV_{DD} / DV_{DD} Fault Indicator – ukazatel indikující, který z komparátorů způsobil poruchu:

1 – komparátor AV_{DD} 0 – komparátor DV_{DD} (u ADuC814 vždy 0)

PSMEN Power Supply Monitor Enable Bit – zapne (log. 1) / vypne (log. 0) monitor napájení

Mikrokontrolér: ADuC824, ADuC834, ADuC836

SFR adresa: DFh

Výchozí hodnota: DEh

Bitové adresace: Ne

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
CMPD	CMPA	PSMI	TPD1	TPD0	TPA1	TPA0	PSMEN

CMPD DV_{DD} Comparator bit – tento bit je jen pro čtení a přímo odráží stav komparátoru napájení DV_{DD} .

1 – napětí DV_{DD} je nad nastavenou mez0 – napětí DV_{DD} je pod nastavenou mez

CMPA AV_{DD} Comparator bit – tento bit je jen pro čtení a přímo odráží stav komparátoru napájení AV_{DD} .

1 – napětí AV_{DD} je nad nastavenou mez0 – napětí AV_{DD} je pod nastavenou mez

PSMI Power Supply Monitor Interrupt Bit – tento bit je nastaven log. 1 pokud je $CMPD = 0$ nebo $CMPA = 0$, pokud $CMPD = 1$ a/nebo $CMPA = 1$ je po 256 ms tento bit nastaven na log. 0 nebo je možné nastavit log. 0 uživatelem.

TPx0 – TPx1 Trip Point Selection Bits – výběr rozhodovací úrovně:

TPx1	TPx0	Rozhodovací úroveň [V]
0	0	4,63
0	1	3,08
1	0	2,93
1	1	2,63

TPDx se vztahují ke komparátoru DV_{DD} a TPAx ke komparátoru AV_{DD} .

PSMEN Power Supply Monitor Enable Bit – zapne (log. 1) / vypne (log. 0) monitor napájení

C.48. PWMCON (PWM control – řídicí slovo PWM)**Mikrokontrolér:** ADuC83x, ADuC84x

SFR adresa: AEh

Výchozí hodnota: 00h

Bitové adresace: Ne

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
SNGL	MD2	MD1	MD0	CDIV1	CDIV0	CSEL1	CSEL0

SNGL Vypne PWM a uvolní výstupy pro klasický IO (vstup/výstupní) režim

MD0-2 PWM Mode Bits – číslo PWM režimu (viz kap. 1.2.4 Pulsně šířkový modulátor PWM)

CDIV0, 1 PWM Clock Divider – dělič základní frekvence

CSEL0, 1 PWM Clock Selector – volba základní frekvence

MD2	MD1	MD0	
0	0	0	Vypnuto
0	0	1	PWM mode 1
0	1	0	PWM mode 2
0	1	1	PWM mode 3
1	0	0	PWM mode 4
1	0	1	PWM mode 5
1	1	0	PWM mode 6
1	1	1	rezervováno

CDIV1	CDIV0	dělič
0	0	1
0	1	4
1	0	16
1	1	64

CSEL1	CSEL0	PWM Clock ¹⁾	PWM Clock ²⁾
0	0	$f_{osc}/D/15^*$	$f_{xtal}/15$
0	1	f_{osc}/D^*	f_{xtal}
1	0	ext T0	ext T0
1	1	f_{osc}	f_{vco}

* D viz koeficient EPM v registru CFG831 nebo CFG841

¹⁾ ADuC831, ADuC841²⁾ ADuC832, ADuC834, ADuC836, ADuC842, ADuC843, ADuC845, ADuC847, ADuC848 $f_{vco}=12,58\text{MHz}$, u ADuC832, ADuC842 a ADuC843 $f_{vco}=16,777216\text{MHz}$ **C.49. PWMxH/L (PWM Data Registers – data do PWM převodníku)****Mikrokontrolér:** ADuC83x, ADuC84x

SFR adresa: B1h (PWM0L), B2h (PWM0H), B3h (PWM1L), B4h (PWM1H)

Výchozí hodnota: 00h (všechny registry)

Bitové adresace: Ne (všechny registry)

Vstupní data pro PWM výstupy. Formát je určen PWM mode (viz kap. 1.2.4 Pulsně šířkový modulátor PWM).

C.50. SEC (Seconds Time Register – registr sekund)**Mikrokontrolér:** ADuC814, ADuC816, ADuC824, ADuC83x, ADuC84x

SFR adresa: A3h

Výchozí hodnota: 00h

Bitová adresace: Ne

Hodnota v tomto registru se zvýší jednou za sekundu při TCEN = 1. SEC čítá od 0 do 59.

C.51. SPICON (SPI Control – řídicí slovo SPI)

Mikrokontrolér: ADuC81x, ADuC824, ADuC834, ADuC836

SFR adresa: F8h

Výchozí hodnota: 04h

Bitové adresace: Ano

bit 7 (FFh)	bit 6 (FEh)	bit 5 (FDh)	bit 4 (FCh)	bit 3 (FBh)	bit 2 (FAh)	bit 1 (F9h)	bit 0 (F8h)
ISPI	WCOL	SPE	SPIM	CPOL	CPHA	SPR1	SPR0

ISPI SPI Interrupt Bit - nastaven na log. 1 na konci každého přenosu SPI, vymazán buď příkazem, nebo vyčtením SPIDAT

WCOL Write Collision Error Bit – nastaven pokud došlo k zápisu do SPIDAT během přenosu. Vymazání jen příkazem.

SPE SPI Interface Enable Bit – volba komunikace SPI / I²C

1 – SPI komunikace

0 – I²C komunikace

SPIM SPI Master/Slave Mode Select Bit – volba režimu SPI komunikace

1 – master mode – SCLOCK je výstup

0 – slave mode – SCLOCK je vstup

CPOL Clock Polarity Select Bit – volba paritního bitu:

1 – vysoká parita – SCLOCK začíná v log. 1

0 – nízká parita – SCLOCK začíná v log. 0

CPHA Clock Phase Select Bit – volba fáze:

1 – náběžná hrana SCLOCK znamená přenos dat

0 – sestupná hrana SCLOCK znamená přenos dat

SPR0, SPR1 SPI Bit Rate Select Bits – volba rychlosti přenosu v master modu

SPR1	SPR0	volba rychlosti
0	0	$f_{CORE}/2$
0	1	$f_{CORE}/4$
1	0	$f_{CORE}/8$
1	1	$f_{CORE}/16$

V režimu slave: SPR0 = vývod SS

C.52. SPIDAT (SPI Data – data SPI komunikace)

Mikrokontrolér: ADuC81x, ADuC824, ADuC834, ADuC836

SFR adresa: F7h

Výchozí hodnota: 00h

Bitové adresace: Ne

Vlastní data přenášená pomocí SPI rozhraní. K těmto datům by se mělo přistupovat jen jednou za cyklus přerušení.

C.53. TIMECON (TIC Control – řídicí slovo TIC)**Mikrokontrolér:** ADuC814, ADuC816, ADuC824, ADuC83x, ADuC84x

SFR adresa: A1h

Výchozí hodnota: 00h

Bitové adresace: Ne

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
-	TFH	ITS1	ITS0	STI	TII	TIEN	TCEN

TFH Twenty-Four Hour Select Bit – dvacet čtyř hodinová volba, pokud je v log. 1, bude počítadlo HOUR počítat od 0 do 23, jinak bude počítat od 0 do 255. (tato funkce podporována jen u ADuC814 a ADuC816 u ostatních je doporučeno z důvodu zpětné kompatibility nastavit log. 1)

INTS0, ITS1 Interval Timebase Selection Bits – volba časové základny pro čítač INTVAL

ITS1	ITS0	časová základna
0	0	1/128 s
0	1	1 s
1	0	1 min
1	1	1 hod

STI Single Time Interval Bit – volba jednoho průběhu – pokud je tento bit nastaven, je po dočasování interního časovače do hodnoty INTVAL shozen příznak TIEN, v opačném případě dojde k opětovnému startu čítání.

TII TIC Interrupt Bit – tento bit je nastaven po dočasování interního čítače do hodnoty INTVAL. Vymazání je nutné provést příkazem.

TIEN Time Interval Enable Bit – pokud je tento bit v log.1, dojde k čítání vnitřního čítače, jinak je hodnota vnitřního čítače nulována.

TCEN Time Clock Enable Bit – pokud je tento bit v log.1, dojde k čítání hodin (HTHSEC, SEC, MIN a HOUR). Do těchto časových registrů je možno zapisovat jen pokud je TCEN = 0.

C.54. WDCON (Watchdog Timer – hlídací časovač)

Mikrokontrolér: ADuC812

SFR adresa: C0h

Výchozí hodnota: 00h

Bitové adresace: Ano

bit 7 (C7h)	bit 6 (C6h)	bit 5 (C5h)	bit 4 (C4h)	bit 3 (C3h)	bit 2 (C2h)	bit 1 (C1h)	bit 0 (C0h)
PRE2	PRE1	PRE0	-	WDR1	WDR2	WDS	WDE

PRE0 – PRE2

Watchdog Timer Prescale Bits – nastavení časového limitu hlídání

PRE2	PRE1	PRE0	časový limit [ms]
0	0	0	16
0	0	1	32
0	1	0	64
0	1	1	128
1	0	0	256
1	0	1	512
1	1	0	1024
1	1	1	2048

WDR1, WDR2

Watchdog Timer Refresh Bits – pro obnovení hlídacího času je potřeba postupně nastavit log. 1 nejprve WDR1 a poté WDR2.

WDS

Watchdog Status Bit – indikátor, že došlo k překročení časového limitu. Vymazat lze příkazem nebo hardwarovým resetem. Zůstane zachován po watchdog resetu.

WDE

Watchdog Enable Bit – zapnutí (log. 1) hlídače,

Mikrokontrolér: ADuC814, ADuC816, ADuC824, ADuC834, ADuC836, ADuC84x

SFR adresa: C0h

Výchozí hodnota: 10h

Bitové adresace: Ano

bit 7 (C7h)	bit 6 (C6h)	bit 5 (C5h)	bit 4 (C4h)	bit 3 (C3h)	bit 2 (C2h)	bit 1 (C1h)	bit 0 (C0h)
PRE3	PRE2	PRE1	PRE0	WDIR	WDS	WDE	WDWR

PRE0 – PRE3

Watchdog Timer Prescale Bits – nastavení časového limitu hlídání
 $t_{WD} = (2^{PRE} \cdot (2^9 / f_{PLL}))$ pro $f_{PLL} = 32,768\text{kHz}$ nebo $32\text{kHz} \pm 10\%$ (u R/C oscilátoru):

PRE3	PRE2	PRE1	PRE0	časový limit [ms]
0	0	0	0	15,6
0	0	0	1	31,2
0	0	1	0	62,5
0	0	1	1	125
0	1	0	0	250
0	1	0	1	500
0	1	1	0	1000
0	1	1	1	2000
1	0	0	0	okamžitý RESET

WDIR

Watchdog Interrupt Request – pokud je bit nastaven na log. 1 provede po dočasování hlídacího limitu přerušení místo resetu systému. Toto přerušení není zakázáno instrukcí CLR EA a má vysokou prioritu. Pokud watchdog není používán k monitorování systému, může být alternativně použit jako časovač.

WDS

Watchdog Status Bit – indikátor, že došlo k překročení časového limitu. Vymazat lze příkazem nebo hardwarovým resetem. Zůstane zachován po watchdog resetu.

WDE

Watchdog Enable Bit – zapnutí (log. 1) hlídače,

WDWR

Watchdog Write Enable Bit – pro obnovení hlídacího času je potřeba dvojitá posloupnost instrukcí. WDWR musí být nastaven a následně provést zápis do WDCON.

Příloha D – Instrukční sada 8051 (8052)

	čerpáno z [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14]
A (Acc)	- akumulátor
Rn	- registr R0-R7 aktuálně vybrané banky registrů
@Ri	- nepřímo adresované paměťové místo pomocí R0 nebo R1
C, AC, OV	- příznaky
PC	- programový čítač
direct	- přímo adresované paměťové místo
@Ri	- nepřímo adresované paměťové místo. 8-bit internal data RAM location (0-255) addressed indirectly through register R1 or R0.
#data	- 8-bitová konstanta
#data16	- 16-bitová konstanta
Addr11	- 11-bitová adresa programové paměti. Používá se u ACALL a AJMP.
Addr16	- 16-bitová adresa programové paměti. Používá se u LCALL a LJMP.
Rel	- relativní adresa. Používá SJMP a všechny podmíněné skoky. Rozsah je -128 až 127 bajtů vzhledem k první bajtu instrukce.
Bit	- přímá adresa bitu v interní paměti nebo v SFR.

Symbol		Mnemonika	
1. operand	Hex	Popis instrukce	Byty
2. operand	Hex		Strojové cykly
3. operand	Hex		Příznaky
ACALL addr11		$PC_{7-0} \rightarrow (SP + 1); PC_{15-8} \rightarrow (SP + 2); Addr11 \rightarrow PC_{10-0}; SP - 2 \rightarrow SP$	
xxx1 0001	11-F1	Nepodmíněné volání podprogramu uvnitř 2kB adresového prostoru	2
Addr ₇₋₀			3
			-
ADD A,Rn		$A + Rn \rightarrow A \quad n = 0 - 7$	
0010 1rrr	28-F	Sčítání obsahu registru Rn a akumulátoru	1
			1
			C, AC, OV, P
ADD A,direct		$A + (direct) \rightarrow A$	
0010 0101	25	Sčítání obsahu přímo adresovaného místa v paměti a akumulátoru	2
direct			2
			C, AC, OV, P
ADD A,@Ri		$A + (Ri) \rightarrow A \quad i = 0, 1$	
0010 011i	26-7	Sčítání obsahu nepřímo adresovaného místa v paměti a akumulátoru	1
			2
			C, AC, OV, P
ADD A,#data		$A + data \rightarrow A$	
0010 0100	24	Sčítání konstanty a akumulátoru	2
data			2
			C, AC, OV, P

Symbol		Mnemonika	
1. operand	Hex	Popis instrukce	Byty
2. operand	Hex		Strojové cykly
3. operand	Hex		Příznaky
ADDC A,Rn		$A + Rn + C \rightarrow A \quad n = 0 - 7$	
0011 1rrr	38-F	Sčítání obsahu registru Rn, bitu přetečení (C) a akumulátoru	1
			1
			C, AC, OV, P
ADDC A,direct		$A + (\text{direct}) + C \rightarrow A$	
0011 0101	35	Sčítání obsahu přímo adresovaného místa v paměti, bitu přetečení (C) a akumulátoru	2
direct			2
			C, AC, OV, P
ADDC A,@Ri		$A + (Ri) + C \rightarrow A \quad i = 0, 1$	
0011 011i	36-7	Sčítání obsahu nepřímo adresovaného místa v paměti, bitu přetečení (C) a akumulátoru	1
			2
			C, AC, OV, P
ADDC A,#data		$A + \text{data} + C \rightarrow A$	
0011 0100	34	Sčítání konstanty, bitu přetečení (C) a akumulátoru	2
data			2
			C, AC, OV, P
AJMP addr11		$\text{Addr11} \rightarrow PC_{10-0}$	
xxx0 0001	01-E1	Skok uvnitř 2kB adresového prostoru	2
Addr ₇₋₀			3
			-
ANL A,Rn		$A \cap Rn \rightarrow A \quad n = 0 - 7$	
0101 1rrr	58-F	Logický součin obsahu registru Rn a akumulátoru	1
			1
			P
ANL A,direct		$A \cap (\text{direct}) \rightarrow A$	
0101 0101	55	Logický součin obsahu přímo adresovaného místa v paměti a akumulátoru	2
direct			2
			P
ANL A,@Ri		$A \cap (Ri) \rightarrow A \quad i = 0, 1$	
0101 011i	56-7	Logický součin obsahu nepřímo adresovaného místa v paměti a akumulátoru	1
			2
			P
ANL A,#data		$A \cap \text{data} \rightarrow A$	
0101 0100	54	Logický součin konstanty a akumulátoru	2
data			2
			P

Symbol		Mnemonika	
1. operand	Hex	Popis instrukce	Byty
2. operand	Hex		Strojové cykly
3. operand	Hex		Příznaky
ANL direct,A		$(\text{direct}) \cap A \rightarrow (\text{direct})$	
0101 0010	52	Logický součin obsahu akumulátoru a přímo adresovaného místa v paměti	2
direct			2
			-
ANL direct,#data		$(\text{direct}) \cap \text{data} \rightarrow (\text{direct})$	
0101 0011	53	Logický součin konstanty a obsahu přímo adresovaného místa v paměti	3
direct			3
data			-
ANL C,bit		$C \cap (\text{bit}) \rightarrow C$	
1000 0010	82	Logický součin bitu přetečení (C) s bitem bitově adresovaného paměťového místa SFR	2
bit			2
			C
ANL C, $\overline{\text{bit}}$		$C \cap (\overline{\text{bit}}) \rightarrow C$	
1011 0000	B0	Logický součin bitu přetečení (C) s negovaným bitem bitově adresovaného paměťového místa SFR	2
bit			2
			C
CJNE A,direct,rel		$A < \text{direct} \rightarrow C = 1 \rightarrow \text{PC} + 3 + \text{Rel} \rightarrow \text{PC}$	
1011 0101	B5	Porovnání obsahu přímo adresovaného místa SFR s obsahem akumulátoru a skok, když nejsou shodné	3
direct			4
rel			C
CJNE A,#data,rel		$A < \text{data} \rightarrow C = 1 \rightarrow \text{PC} + 3 + \text{Rel} \rightarrow \text{PC}$	
1011 0100	B4	Porovnání konstanty s obsahem akumulátoru a skok, když nejsou shodné	3
data			4
rel			C
CJNE Rn,#data,rel		$R_n < \text{data} \rightarrow C = 1 \rightarrow \text{PC} + 3 + \text{Rel} \rightarrow \text{PC}$	
1011 1rrr	B8-F	Porovnání konstanty s obsahem registru Rn a skok, když nejsou shodné	3
data			4
rel			C
CJNE @Ri,#data,rel		$(R_i) < \text{data} \rightarrow C = 1 \rightarrow \text{PC} + 3 + \text{Rel} \rightarrow \text{PC}$	
1011 011i	B6-7	Porovnání konstanty s obsahem nepřímo adresovaného paměťového místa a krátký relativní skok, když nejsou shodné	3
data			4
rel			C
CLRA		$0 \rightarrow A$	
1110 0100	E4	Nulování obsahu akumulátoru	1
			1
			P

Symbol		Mnemonika	
1. operand	Hex	Popis instrukce	Byty
2. operand	Hex		Strojové cykly
3. operand	Hex		Příznaky
CLR C		$0 \rightarrow C$	
1100 0011	C3	Nulování obsahu bitu přetečení (C)	1
			1
			C
CLR bit		$0 \rightarrow (\text{bit})$	
1100 0010	C2	Nulování obsahu bitově adresovaného SFR	2
bit			2
			-
CPL A		$\bar{A} \rightarrow A$	
1111 0100	F4	Negace obsahu akumulátoru	1
			1
			-
CPL C		$\bar{C} \rightarrow C$	
1011 0011	B3	Negace obsahu bitu přetečení (C)	1
			1
			C
CPL bit		$(\bar{\text{bit}}) \rightarrow (\text{bit})$	
1011 0010	B2	Negace obsahu bitově adresovaného SFR	2
bit			2
			-
DAA			
1101 0100	D4	Desítková korekce obsahu akumulátoru	1
			2
			C, P
DEC A		$A - 1 \rightarrow A$	
0001 0100	14	Odečtení 1 od obsahu akumulátoru	1
			1
			P
DEC Rn		$R_n - 1 \rightarrow R_n \quad n = 0 - 7$	
0001 1rrr	18-F	Odečtení 1 od obsahu registru Rn	1
			1
			-
DEC direct		$(\text{direct}) - 1 \rightarrow (\text{direct})$	
0001 0101	15	Odečtení 1 od obsahu přímo adresovaného místa SFR	2
direct			2
			-

Symbol		Mnemonika	
1. operand	Hex	Popis instrukce	Byty
2. operand	Hex		Strojové cykly
3. operand	Hex		Příznaky
DEC @Ri		$(Ri) - 1 \rightarrow (Ri)$	
0001 011i	16-7	Odečtení 1 od obsahu nepřímě adresovaného místa v paměti	1
			2
			-
DIV AB		$A / B \rightarrow A$	
1000 0100	84	Dělení obsahu akumulátoru a registru B	1
			9
			C, OV, P
DJNZ Rn,rel		$Rn - 1 \rightarrow Rn ; Rn \neq 0 \rightarrow PC + 2 + Rel \rightarrow PC$	
1101 1rrr	D8-F	Odečtení 1 od obsahu registru Rn a krátký relativní skok, když není nulový	2
rel			3
			-
DJNZ direct,rel		$(direct) - 1 \rightarrow (direct) ; (direct) \neq 0 \rightarrow PC + 3 + Rel \rightarrow PC$	
1101 0101	D5	Odečtení 1 od obsahu přímo adresovaného místa SFR a krátký relativní skok, když není nulový	3
direct			4
rel			-
INC A		$A + 1 \rightarrow A$	
0000 0100	04	Přičtení 1 k obsahu akumulátoru	1
			1
			P
INC Rn		$Rn + 1 \rightarrow Rn \quad n = 0 - 7$	
0000 1rrr	08-F	Přičtení 1 k obsahu registru Rn	1
			1
			-
INC direct		$(direct) + 1 \rightarrow (direct)$	
0000 0101	05	Přičtení 1 k obsahu přímo adresovaného místa v paměti	2
direct			2
			-
INC @Ri		$(Ri) + 1 \rightarrow (Ri) \quad i = 0, 1$	
0000 011i	06-7	Přičtení 1 k obsahu nepřímě adresovaného místa v paměti	1
			2
			-
INC DPTR		$DPTR + 1 \rightarrow DPTR$	
1010 0011	A3	Přičtení 1 k obsahu ukazatele DPTR	1
			3
			-

Symbol		Mnemonika	
1. operand	Hex	Popis instrukce	Byty
2. operand	Hex		Strojové cykly
3. operand	Hex		Příznaky
JB bit, rel		(bit) = 1 → PC +3 + rel → PC	
0010 0000	20	Pokud je obsah v bitově adresovaného SFR roven logické 1, provede se krátký relativní skok	3
bit			4
rel			-
JBC bit, rel		(bit) = 1 → 0 → (bit) ; PC +3 + rel → PC	
0001 0000	10	Pokud je obsah v bitově adresovaného SFR roven logické 1, obsah se vynuluje a provede se krátký relativní skok	3
bit			4
rel			-
JC rel		C = 1 → PC +2 + rel → PC	
0100 0000	40	Pokud je obsah bitu přetečení (C) roven logické 1, provede se krátký relativní skok	2
rel			2
			-
JMP @A+DPTR		A + DPTR → PC	
0111 0011	73	Skok na adresu, která je určena sčítáním obsahu akumulátoru a DPTR	1
			3
			-
JNB bit, rel		(bit) = 0 → PC +3 + rel → PC	
0011 0000	30	Pokud je obsah v bitově adresovaného SFR roven logické 0, provede se krátký relativní skok	3
bit			4
rel			-
JNC rel		C = 0 → PC +2 + rel → PC	
0101 0000	50	Pokud je obsah bitu přetečení (C) roven logické 0, provede se krátký relativní skok	2
rel			3
			-
JNZ rel		A ≠ 0 → PC +2 + rel → PC	
0111 0000	70	Pokud není obsah akumulátoru roven 0, provede se krátký relativní skok	2
rel			3
			-
JZ rel		A = 0 → PC +2 + rel → PC	
0110 0000	60	Pokud je obsah akumulátoru roven 0, provede se krátký relativní skok	2
rel			3
			-
LCALL addr16		PC ₇₋₀ → (SP + 1) ; PC ₁₅₋₈ → (SP + 2) ; Addr16 → PC ; SP + 2 → SP	
0001 0010	12	Nepodmíněné volání podprogramu uvnitř 64kB adresového prostoru	3
Addr ₇₋₀			4
Addr ₁₅₋₈			-

Symbol		Mnemonika	
1. operand	Hex	Popis instrukce	Byty
2. operand	Hex		Strojové cykly
3. operand	Hex		Příznaky
LJMP addr16		Addr16 → PC	
0000 0010	02	Skok uvnitř 64kB adresového prostoru	3
Addr ₇₋₀			4
Addr ₁₅₋₈			-
MOV A,Rn		Rn → A n = 0 - 7	
1110 1rrr	E8-F	Přesun obsahu registru Rn do akumulátoru	1
			1
			P
MOV A,direct		(direct) → A	
1110 0101	E5	Přesun obsahu přímo adresovaného místa v paměti do akumulátoru	2
direct			2
			P
MOV A,@Ri		(Ri) → A i = 0, 1	
1110 011i	E6-7	Přesun obsahu nepřímo adresovaného místa v paměti do akumulátoru	1
			2
			P
MOV A,#data		data → A	
0111 0100	74	Přesun konstanty do akumulátoru	2
data			2
			P
MOV Rn,A		A → Rn n = 0 - 7	
1111 1rrr	F8-F	Přesun obsahu akumulátoru do registru Rn	1
			1
			-
MOV Rn,direct		(direct) → Rn n = 0 - 7	
1010 1rrr	A8-F	Přesun obsahu přímo adresovaného místa v paměti do registru Rn	2
direct			2
			-
MOV Rn,#data		data → Rn n = 0 - 7	
0111 1rrr	78-F	Přesun konstanty do registru Rn	2
data			2
			-
MOV direct, A		A → (direct)	
1111 0101	F5	Přesun obsahu akumulátoru do přímo adresovaného místa v paměti	2
direct			2
			-

Symbol		Mnemonika	
1. operand	Hex	Popis instrukce	Byty
2. operand	Hex		Strojové cykly
3. operand	Hex		Příznaky
MOV direct, Rn		Rn → (direct) n = 0 - 7	
1000 1rrr	88-F	Přesun obsahu registru Rn do přímo adresovaného místa v paměti	2
direct			2
			-
MOV direct, direct		(direct) → (direct)	
1000 0101	85	Přesun obsahu přímo adresovaného místa v paměti do přímo adresovaného místa v paměti	3
direct			3
direct			-
MOV direct,@Ri		(Ri) → (direct) i = 0, 1	
1000 011i	86-7	Přesun obsahu nepřímo adresovaného místa v paměti do přímo adresovaného místa v paměti	2
direct			2
			-
MOV direct,#data		data → (direct)	
0111 0101	75	Přesun konstanty do přímo adresovaného místa v paměti	3
direct			3
data			-
MOV @Ri,A		A → (Ri) i = 0, 1	
1111 011i	F6-7	Přesun obsahu akumulátoru do nepřímo adresovaného místa v paměti	1
			2
			-
MOV @Ri,direct		(direct) → (Ri) i = 0, 1	
1010 011i	A6-7	Přesun obsahu přímo adresovaného místa v paměti do nepřímo adresovaného místa v paměti	2
direct			2
			-
MOV @Ri,#data		data → (Ri) i = 0, 1	
0111 011i	76-7	Přesun konstanty do nepřímo adresovaného místa v paměti	2
data			2
			-
MOV C,bit		(bit) → C	
1010 0010	A2	Přesun obsahu v bitově adresovaného SFR do bitu přetečení (C)	2
bit			2
			C
MOV direct, C		C → (bit)	
1001 0010	92	Přesun obsahu bitu přetečení (C) do bitově adresovaného SFR	2
bit			2
			-

Symbol		Mnemonika	
1. operand	Hex	Popis instrukce	Byty
2. operand	Hex		Strojové cykly
3. operand	Hex		Příznaky
MOV DPTR,#data16		Data16 → DPTR	
1001 0000	90	Přesun 16-bitové konstanty do ukazatele DPTR	3
Data ₁₅₋₈			3
Data ₇₋₀			-
MOVC A,@A+DPTR		(A + DPTR) → A	
1001 0011	93	Přesun do akumulátoru z přímo adresovaného místa v paměti, kde adresa je určena sčítáním obsahu akumulátoru a DPTR	1
			4
			P
MOVC A,@A+PC		(A + PC) → A	
1000 0011	83	Přesun do akumulátoru z přímo adresovaného místa v paměti, kde adresa je určena sčítáním obsahu akumulátoru a PC	1
			4
			P
MOVX A,@Ri		(Ri) → A i = 0, 1	
1110 001i	E2-3	Přesun obsahu nepřímo adresovaného místa v externí paměti do akumulátoru	1
			4
			P
MOVX A,@DPTR		(DPTR) → A	
1110 0000	E0	Přesun obsahu nepřímo adresovaného místa v externí paměti do akumulátoru	1
			4
			P
MOVX @Ri,A		A → (Ri) i = 0, 1	
1111 001i	F2-3	Přesun obsahu akumulátoru do nepřímo adresovaného místa v externí paměti	1
			4
			-
MOVX @DPTR,A		A → (DPTR)	
1111 0000	F0	Přesun obsahu akumulátoru do nepřímo adresovaného místa v externí paměti	1
			4
			-
MUL AB		A × B → A	
1010 0100	A4	Násobení obsahu akumulátoru a registru B	1
			9
			C, OV, P
NOP			
0000 0000	00	Prázdná instrukce	1
			1
			-

Symbol		Mnemonika	
1. operand	Hex	Popis instrukce	Byty
2. operand	Hex		Strojové cykly
3. operand	Hex		Příznaky
ORL A,Rn		$A \cup R_n \rightarrow A \quad n = 0 - 7$	
0100 1rrr	48-F	Logický součet obsahu registru Rn a akumulátoru	1
			1
			P
ORL A,direct		$A \cup (\text{direct}) \rightarrow A$	
0100 0101	45	Logický součet obsahu přímo adresovaného místa v paměti a akumulátoru	2
direct			2
			P
ORL A,@Ri		$A \cup (R_i) \rightarrow A \quad i = 0, 1$	
0100 011i	46-7	Logický součet obsahu nepřímo adresovaného místa v paměti a akumulátoru	1
			2
			P
ORL A,#data		$A \cup \text{data} \rightarrow A$	
0100 0100	44	Logický součet konstanty a akumulátoru	2
data			2
			-
ORL direct,A		$(\text{direct}) \cup A \rightarrow (\text{direct})$	
0100 0010	42	Logický součet obsahu akumulátoru a přímo adresovaného místa v paměti	2
direct			2
			-
ORL direct,#data		$(\text{direct}) \cup \text{data} \rightarrow (\text{direct})$	
0100 0011	43	Logický součet konstanty a obsahu přímo adresovaného místa v paměti	3
direct			3
data			-
ORL C,bit		$C \cup (\text{bit}) \rightarrow C$	
0111 0010	72	Logický součet obsahu bitově adresovaného SFR a bitu přetečení (C)	2
bit			2
			C
ORL C,bit		$C \cup (\overline{\text{bit}}) \rightarrow C$	
1010 0000	A0	Logický součet negace obsahu bitově adresovaného SFR a bitu přetečení (C)	2
bit			2
			C
POP direct		$(SP) \rightarrow (\text{direct}); SP - 1 \rightarrow SP$	
1101 0000	D0	Přesun obsahu zásobníku do přímo adresovaného místa v paměti	2
direct			2
			-

Symbol		Mnemonika	
1. operand	Hex	Popis instrukce	Byty
2. operand	Hex		Strojové cykly
3. operand	Hex		Příznaky
PUSH direct		SP + 1 → SP ; (direct) →(SP)	
1100 0000	C0	Přesun obsahu přímo adresovaného místa v paměti do zásobníku	2
direct			2
			-
RET		(SP) → PC ₁₅₋₈ ; (SP - 1) → PC ₇₋₀ ; SP - 2 → SP	
0010 0010	22	Návrat z podprogramu	1
			4
			-
RETI		(SP) → PC ₁₅₋₈ ; (SP - 1) → PC ₇₋₀ ; SP - 2 → SP	
0011 0010	32	Návrat z obslužného podprogramu přerušení	1
			4
			-
RLA		A _{n+1} → A _n kde n = 0 – 6 ; A ₀ → A ₇	
0010 0011	23	Rotace obsahu akumulátoru o jednu pozici vlevo	1
			1
			-
RLCA		A _{n+1} → A _n kde n = 0 – 6 ; A ₀ → C ; C → A ₇	
0011 0011	33	Rotace obsahu akumulátoru o jednu pozici vlevo přes carry bit	1
			1
			C, P
RRA		A _n → A _{n+1} kde n = 0 – 6 ; A ₇ → A ₀	
0000 0011	03	Rotace obsahu akumulátoru o jednu pozici vpravo	1
			1
			-
RRC A		A _n → A _{n+1} kde n = 0 – 6 ; A ₇ → C ; C → A ₀	
0001 0011	13	Rotace obsahu akumulátoru o jednu pozici vpravo přes carry bit	1
			1
			C, P
SETB C		1 → C	
1101 0011	D3	Nastavení log.1 obsahu bitu přetečení (C)	1
			1
			C
SETB bit		1 → (bit)	
1101 0010	D2	Nastavení log.1 obsahu bitově adresovaného SFR	2
bit			2
			-

Symbol		Mnemonika	
1. operand	Hex	Popis instrukce	Byty
2. operand	Hex		Strojové cykly
3. operand	Hex		Příznaky
SJMP rel		$PC + 2 + rel \rightarrow PC$	
1000 0000	80	Provede se krátký relativní skok	2
rel			3
			-
SUBB A,Rn		$A - Rn - C \rightarrow A \quad n = 0 - 7$	
1001 1rrr	98-F	Odečtení obsahu registru Rn od akumulátoru a obsahu bitu přetečení (C)	1
			1
			C, AC, OV
SUBB A,direct		$A - (direct) - C \rightarrow A$	
1001 0101	95	Odečtení obsahu přímo adresovaného místa v paměti od akumulátoru a obsahu bitu přetečení (C)	2
direct			2
			C, AC, OV
SUBB A,@Ri		$A - (Ri) - C \rightarrow A \quad i = 0, 1$	
1001 011i	96-7	Odečtení obsahu nepřímo adresovaného místa v paměti od akumulátoru a obsahu bitu přetečení (C)	1
			2
			C, AC, OV
SUBB A,#data		$A - data - C \rightarrow A$	
1001 0100	94	Odečtení konstanty od akumulátoru a obsahu bitu přetečení (C)	2
data			2
			C, AC, OV
SWAP A		$A_{0-3} \leftrightarrow A_{4-7}$	
1100 0100	C4	Výměna vyšších čtyř bitů s nižšími čtyřmi bity	1
			1
			-
XCH A,Rn		$Rn \leftrightarrow A \quad n = 0 - 7$	
1100 1rrr	C8-F	Výměna obsahu registru Rn a akumulátoru	1
			1
			-
XCH A,direct		$(direct) \leftrightarrow A$	
1100 0101	C5	Výměna obsahu přímo adresovaného místa v paměti a akumulátoru	2
direct			2
			-
XCH A,@Ri		$(Ri) \leftrightarrow A \quad i = 0, 1$	
1100 011i	C6-7	Výměna obsahu nepřímo adresovaného místa v paměti a akumulátoru	1
			2
			-

Symbol		Mnemonika	
1. operand	Hex	Popis instrukce	Byty
2. operand	Hex		Strojové cykly
3. operand	Hex		Příznaky
XCHD A,@Ri		$(Ri_{0-3}) \leftrightarrow A_{0-3} \quad i = 0, 1$	
1101 011i	D6-7	Výměna nižších čtyř bitů obsahu nepřímě adresovaného místa v paměti s nižšími čtyřmi bity akumulátoru	1
			2
			-
XRL A,Rn		$A \oplus Rn \rightarrow A \quad n = 0 - 7$	
0110 1rrr	68-F	Logické výhradní nebo „XOR“ obsahu registru Rn a akumulátoru	1
			1
			-
XRL A,direct		$A \oplus (\text{direct}) \rightarrow A$	
0110 0101	65	Logické výhradní nebo „XOR“ obsahu přímo adresovaného místa v paměti a akumulátoru	2
direct			2
			-
XRL A,@Ri		$A \oplus (Ri) \rightarrow A \quad i = 0, 1$	
0110 011i	66-7	Logické výhradní nebo „XOR“ obsahu nepřímě adresovaného místa v paměti a akumulátoru	1
			2
			-
XRL A,#data		$A \oplus \text{data} \rightarrow A$	
0110 0100	64	Logické výhradní nebo „XOR“ konstanty a obsahu akumulátoru	2
data			2
			-
XRL direct,A		$(\text{direct}) \oplus A \rightarrow (\text{direct})$	
0110 0010	62	Logické výhradní nebo „XOR“ obsahu akumulátoru a přímo adresovaného místa v paměti	2
direct			2
			-
XRL direct,#data		$(\text{direct}) \oplus \text{data} \rightarrow (\text{direct})$	
0110 0011	63	Logické výhradní nebo „XOR“ konstanty a obsahu přímo adresovaného místa v paměti	3
direct			3
data			-