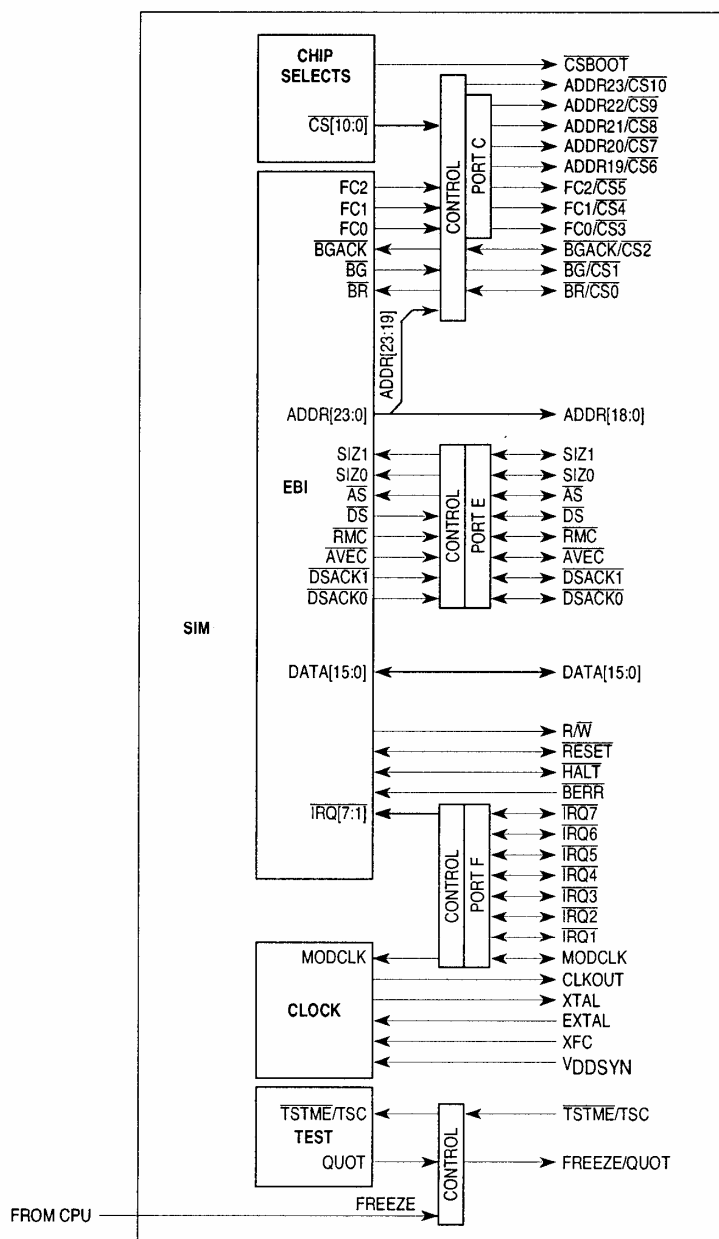


5. MODUL ZA INTEGRACIJO SISTEMA - SIM

SIM (angl. System Integration Module - modul za integracijo sistema) je modul mikrokontroler MC68332, katerega glavne naloge so [24]:

- povezovanje s periferijo,
- generiranje urinega takta za ostale module,
- sistemska zaščita,
- krmiljenje in zajemanje vhodno/izhodnih digitalnih signalov
- generiranje signalov za izbiro čipov.



Slika 5. 1: Blokovna shema SIM z nožicami

Funkcionalni bloki SIM so (slika 5. 1):

- **konfiguracija sistema in zaščita** (angl. System Configuration and Protection),
- **sintetizator ure** (angl. Clock Synthesizer),
- **preizkušanje sistema** (angl. System Test),
- **vmesnik k zunanjem vodilu** (angl. External Bus Interface - EBI),
- **izbira integriranih vezij - čipov** (angl. chip selects - CS).

Večina nožic SIM je večnamenskih (npr. nožico IRQ1 lahko uporabljamo kot vhod za hardversko prekinitvev ali pa kot navadni binarni vhod ali izhod). Pred zagonom programa je za izbiro funkcije in nastavljanje parametrov SIM treba v ustrezna stanja postaviti ("maskirati") bite na vnaprej določenih spominskih lokacijah. Čeprav gre v tem primeru za RAM pomnilnik in ne za prave registre (kot so npr. podatkovni in naslovni registri CPU), imenujemo te 16-bitne lokacije "SIM registri". Njihove naslove ter nazive kaže slika 5. 2¹.

	15	8	7	0
Yffa00	MODULE CONFIGURATION (MCR)			
Yffa02	MODULE TEST (SIMTR)			
Yffa04	CLOCK SYNTHESIZER CONTROL (SYNCR)			
Yffa06	PRAZNO		RESET STATUS (RSR)	
Yffa08	MODUL TEST E (SIMTRE)			
...	PRAZNO			
Yffa10	PRAZNO		PORTE DATA (PORTE)	
Yffa12	PRAZNO		PORTE DATA (PORTE)	
Yffa14	PRAZNO		PORTE DATA DIRECTION (DDRE)	
Yffa16	PRAZNO		PORTE PIN ASIGNMENT (PEPAR)	
...	enako za PORTF			
Yffa20	PRAZNO		SYSTEM PROTECTION CONTROL (SYPCR)	
Yffa22	PERIODIC INTERRUPT CONTROL (PICR)			
Yffa24	PERIODIC INTERRUPT TIMING (PITR)			
	PRAZNO			
Yffa26	PRAZNO		SOFTWARE SERVICE (SWSR)	
Yffa28	PRAZNO			
...			
Yffa44	CHIP SELECT PIN ASSIGNMENT (CSPAR0)			
Yffa46	CHIP SELECT PIN ASSIGNMENT (CSPAR1)			
Yffa48	CHIP SELECT BASE BOOT (CSBARBT)			
Yffa4a	CHIP SELECT OPTION BOOT (CSORBT)			
...	... registri ostalih <u>CS</u> ...			
Yffa58	CHIP SELECT BASE 3 (CSBAR3)			
Yffa5a	CHIP SELECT OPTION 3 (CSOR3)			
...	... registri ostalih <u>CS</u> ...			
Yffa4a	PRAZNO			

Slika 5. 2: RAM lokacije in nazivi SIM registrov

¹ Prva heksadecimalna številka Y v naslovu je lahko 7_{HEX} ali F_{HEX}, odvisno od bita MM v MCR registru.

Nimamo namena podrobno opisati posamezne registre. Omejili se bomo le na nekatere značilnosti SIM.

5.1 Podmodul za konfiguracijo sistema

Podmodul za **konfiguracijo sistema in zaščito** skrbi za osnovno prednastavitev SIM. V ta namen v registru za konfiguracijo modula - MCR (angl. Module Configuration Register) na primer določimo, ali bo prva številka iz registrske mape SIM 7_{HEX} ali F_{HEX} . Posebno polje določa prioriteto prekinitve SIM modula v primerjavi z zahtevami po prekinitvah ostalih modulov MC68332 itd.

V SYPCR registru (angl. System Protection Control Register) lahko omogočimo in parametriramo čuvaju časa (angl. *watchdog* - dobesedno: pes čuvaj). Osnovna funkcija watchdoga je nadzor nad trajanjem izvajanja programa. Namreč, velika večina programov za delo v realnem času (npr. regulacija motorjev), se izvaja ciklično (običajno kot posledica časovne prekinitve, glej pogl. 10). Celotna programska sekvenca se mora končati preden poteče nek vzorčni interval T_V (glej tudi [2]). Eden od možnih vzrokov za “podaljšanje” izvajanja te sekvence je neskončna zanka, ki je posledica nepravilno zasnovanih internih skokov. Zato je koristno predvideti nadzor nad časom izvajanja programa.

Watchdog poženemo v SYPCR registru, kjer tudi določimo trajanje nadzornega cikla (od $30,6 \mu\text{s}$ do 1 s pri 16,77 MHz osnovnem taktu). Po zagonu watchdoga je treba register SWSR (angl. Software Service Register) periodično (običajno na koncu cikla uporabniškega programa) postaviti v določeno stanje, ki sproži naslednje odštevanje časa. Izostanek te procedure pomeni, da je izvajanje programa daljše od časa, predvidenega v watchdog časovniku, nakar ta povzroči RESET prekinitiv.

Poleg omenjene funkcije lahko v podmodulu za konfiguracijo sprožimo časovnik za periodično prekinitiv (angl. periodic interrupt timer). Posledica je skok v prekinitveni podprogram v pravilnih časovnih intervalih, zato lahko to funkcijo uporabljamo za zagotavljanje intervala vzorčenja. Pomanjkljivost tega pristopa je v relativno grobi izbiri možnega intervala med dvema prekinitvama (od $122 \mu\text{s}$ do 15,9 s z minimalnim intervalom $122 \mu\text{s}$ pri 16,77 MHz urinem taktu; npr. prekinitve s periodo 1 ms ni mogoče realizirati). Zato za generiranje časovne prekinitve raje uporabljamo PWM funkcijo TPU (pogl. 7).

5.2 Sintetizator ure

Frekvence sistemske ure za MC68332 so (odvisno od verzije) 16,77 MHz, 20,97 MHz ali 25 MHz (glej poglavje 4 o CPU). *Kristal*, dajalnik referenčnega takta za uro, pulzira s bistveno nižjo frekvenco; pri prvi verziji se lahko giblje med 25 kHz in 50 kHz (nazivna vrednost 32,768 kHz)! **Sintetizator ure** skrbi za nastavljanje sistemske frekvence prek maskiranja ustreznih bitov v registru SYNCR (angl. Clock Synthesizer Control Register). Na ta način je možna fleksibilna softverska izbira sistemske frekvence, ki daje osnovni takt vsem operacijam mikrokrmilnika! Pri frekvenci kristala 32,768 kHz je omogočena nastavitev sistemske ure prve inačice od 131 kHz do 16,777 MHz. Referenčni takt pa lahko zajemamo tudi iz zunanjega vira.

5.3 Preizkušanje sistema

Namen podmodula za **preizkušanje sistema** je tovarniško testiranje delovanja mikrokrmilnika, zato je uporabniku nedostopen.

5.4 Vmesnik k zunanjem vodilu

5.4.1 Povezava s perifernim vodilom

Na sliki 5. 1 so prikazane vhodno/izhodne nožice vmesnika k zunanjem vodilu. Lahko jih razdelimo v dve osnovni skupini: **zunanje naslovno, krmilno in podatkovno vodilo s hardverskimi prekinitvami** ter **nožice za izbiro integriranih vezij (čipov)**. Slednje bomo obravnavali v posebnem podpoglavju.

Zunanje naslovno, podatkovno in krmilno vodilo rabimo za naslavljanje zunanjih enot (vezij) kot so pomnilniki, A/D in D/A pretvorniki in ostala vezja. Zaradi pomembnosti bomo nožice za hardverske prekinitve obravnavali ločeno.

Celotno **naslovno vodilo** MC68332 obsega 24 nožic (A0-A23), kar pomeni možnost naslavljanja $2^{24} = 16\,777\,216$ zlogov (16 Mbyte^2) perifernih enot. V tem segmentu je zajetih le prvih 19 naslovnih bitov (A0-A18), preostalih pet (A19-A23) pa sodi k modulu za izbiro čipov (glej podpoglavje 5.5).

Interno **podatkovno vodilo** (IMB) omogoča hkratni transfer 32-bitnega podatka med podmoduli v enem samem ciklu. Po drugi strani obsega zunanje vodilo le 16 bitov (D0 - D15). To pomeni, da moramo 32-bitni podatek v/iz periferije poslati/sprejeti v dveh obrokih. Hitrost prenosa podatkov v ali iz zunanjih pomnilnikov je manjša od hitrosti internega prenosa znotraj mikroračunalnika. Obe dejstvi sta razlog za to, da se izkušeni programerji trudijo čim manj komunicirati z zunanjimi pomnilniki. Prav v ta namen poskušajo projektanti procesorjev zagotoviti veliko število internih podatkovnih registrov (pri nas jih imamo osem: D0 - D7). Če vmesnega rezultata neke operacije ne rabimo takoj po njenem izvajanju, ampak šele čez nekaj ukazov, raje ta podatek ohranimo v registru iz katerega ga po potrebi pokličemo. Seveda zahteva takšen pristop premišljeno pisanje programa, ker je tak register do nadaljnjega zaseden in neuporaben za druge operacije³.

Prilagoditev hitrosti transferov med hitrim CPU in počasnejšim pomnilnikom dosežemo z uvajanjem *čakalnih stanj* (angl. wait states).

Krmilno vodilo (angl. control bus) je skorajda v celoti zbrano v *vratih* (angl. port) E (slika 5. 1). Ta imajo dvojno vlogo: njihove nožice lahko delujejo v funkciji krmilnega vodila ali pa kot navadnih osem binarnih vhodov ali izhodov. Za izbiro funkcije moramo maskirati registre PEPAR, DDRE in PORTE (slika 5. 2).

² **PAZI!** 1 Kbyte ni 1000, temveč 1024 bajtov.

³ Med ostalim je tudi to prednost uporabe strojnega jezika pred C-jem, saj slednji, zlasti pri skokih na podprograme, shranjuje vmesne podatke v RAM.

Kot primer lahko omenimo delovanje izhodov SIZ1 in SIZ0 ter vhoda DSACK0 in DSACK1, ki skupaj z bitom naslovnega vodila A0 sinhronizirajo komunikacijo s periferijo. Pogosto se namreč dogaja, da je nek podatek treba poslati ali prečitati v/iz 8- ali 16-bitne periferne enote. Ena od nalog omenjenih signalov je prilagoditev 32-bitnega ali 8-bitnega podatka iz CPU na enoto z drugačno dolžino vodila, enako kot izbira pozicije poslanega byta (soda ali liha). Pri tem pa izhoda AS in DS (angl. address strobe, data strobe) signalizirata aktivnost naslovnega oz. podatkovnega vodila.

5.4.2 Digitalne vhodno/izhodne nožice

Ena od glavnih značilnosti mikrokrmilnika MC68332 je možnost izbire funkcije nožic. Večina nožic v SIM in QSM ima vsaj dvojno funkcijo; poleg specifičnih funkcij (vodila, CS nožice, hardverske prekinitve...) jih lahko uporabljamo kot splošne digitalne vhodno/izhodne nožice. Ti priključki so zbrani v vratih E, F in C (z določenimi omejitvami). Po izbiri funkcije nožice (PFPAR in PEPAR: 1 - standardna funkcija s slike 5. 1, 0 - vhodno/izhodna funkcija) izberemo smer digitalnih signalov (0 - izhod, 1 - vhod) v registrih DDRE in DDRF. V registrih PORTE0 in PORTE1 in PORTF0 in PORTF1 določimo vrednost izhodnega signala ali preberemo vrednost vhodnega signala (slika 5. 2).

5.4.3 Hardverske prekinitve

Sedem nožic na portu F lahko uporabljamo kot vhode za zunanje hardverske prekinitve (IRQ1-IRQ7; angl. interrupt request - zahteva po prekinitvi). Vhodi imajo že vnaprej določeno prioriteto: ob hkrantni zahtevi po več zunanjih prekinitvah ima IRQ7 največjo, IRQ1 pa najmanjšo prioriteto. Vse prekinitve razen IRQ7 lahko maskiramo v statusnem registru CPU (glej tudi pogl. 4.2.8.1.1). Če je prekinitve odobrena, pride do izvajanja prekinitvenega programa za ustrezni IRQ. Naslovi prekinitvenih podprogramov se nahajajo v vektorski tabeli na lokacijah 64_{HEX} - 7C_{HEX}.

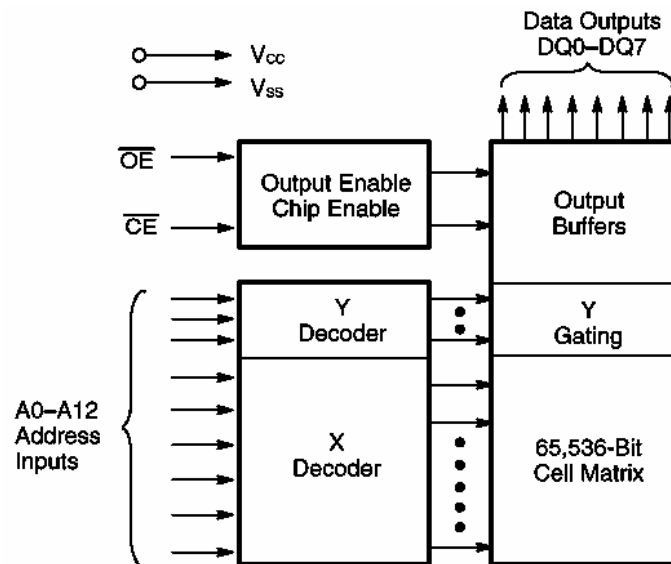
5.5 Izbira integriranih vezij

5.5.1 Problematika naslavljanja čipov

24-bitno naslovno vodilo omogoča naslavljanje 16 M (000000_{HEX} - FFFFFFF_{HEX}) bytnih lokacij. Znotraj tega intervala so nekatera področja rezervirana za različne registre podmodulov ali pa za inicializacijski program ter delovni RAM (glej pogl. 8). Ostala nerezervirana področja lahko uporabimo za naslavljanje zunanjih enot (RAM in ROM pomnilnikov, pretvornikov, prikazovalnikov itd.). Pri snovanju arhitekture našega mikroračunalniškega sistema moramo seveda poskrbeti za to, da so naslovi zunanjih enot enoumno določeni, torej dve lokaciji ne smeta imeti istega naslova.

Oglejmo si primer naslavljanja 8 Kbyte RAM pomnilnika (8192_{DEC} = 2000_{HEX} lokacij velikosti enega byta), ki je prikazan na sliki 5. 3. Pomnilnik želimo postaviti na prosto področje F00000_{HEX} - F01FFF_{HEX}. Na vezju so za to zadolžene naslednje nožice (podobno velja tudi za ostala vezja):

- naslovne nožice za priključitev na naslovno vodilo mikoračunalnika (A12 - A0),
- nožica za izbiro vezja CS.



Slika 5. 3: Blokovna shema pomnilnika AMD27X64 (Advanced Micro Devices) velikosti 8 Kbyte x 8-bit

Navedeni interval naslovov pomnilnika lahko zapišemo tudi v binarni obliki

1111 0000 0000 0000 0000 000_{BIN} - 1111 0000 0001 1111 1111 1111_{BIN},

kjer se vsako naslavljanje kombinacije

1111 0000 000X XXXX XXXX XXXX_{BIN}⁴

nanaša na neko lokacijo znotraj izbranega RAM. Skrajnje levi bit (MSB) ustreza naslovnemu bitu A23, skrajnji desni (LSB) pa A0.

Iz opisanega lahko ugotovimo naslednje:

1. Biti, označeni z X, določajo naslove v intervalu 8 K znotraj čipa (lokalni naslovi).
2. Preostala kombinacija bitov določa absolutno lokacijo čipa, to je položaj teh 8K v spominski mapi MC68332.

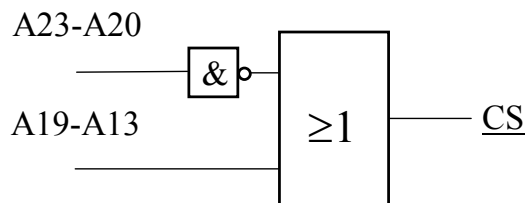
Ustrezno temu sledijo praktični napotki:

1. X bite moramo vezati neposredno na naslovne nožice vezja A12 - A0 ($2^{13} = 8192_{DEC} = 8 K$).

⁴ X pomeni logično stanje 0 ali 1.

2. Čip bo naslovljen le, če se bo na naslovnih nožicah A23 - A13 mikrokrmilnika pojavila kombinacija 1111 0000 000, torej je treba omogočiti, da se ob njej aktivira signal CS (aktiven na logično 0).

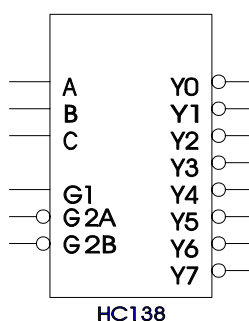
Najprimitivnejše vezje za izbiro čipa je izrisano na sliki 5. 4.



Slika 5. 4: Principialna logična shema vezja za omogočanje CS

Za realizacijo vezja s prejšnje slike bi rabili nekaj čipov: naslovne bite A23 - A20 je treba povezati preko logične funkcije NEIN (angl. NAND), njen izhod pa skupaj z ostalimi biti A19 - A13 pripeljemo na logično funkcijo ALI (angl. OR). Takrat se le ob izbrani kombinaciji bitov na izhodu pojavi logična 0. Za realizacijo opisanega pristopa bi potrebovali precej čipov. Če temu dodamo še dejstvo, da je treba podobna vezja realizirati tudi za izbiro ostalih perifernih integriranih vezij, kmalu ugotovimo, da bi rabili ogromno čipov z logičnimi funkcijami, kar bi pomenilo veliko "gnečo" na tiskanem vezju.

Omenimo še dodatni problem naslavljanja čipov, ki zasedata spominsko področje le enega ali dveh bytov. Takšni so npr. A/D in D/A pretvorniki (imajo tudi CS vhod). 12-bitni pretvornik moramo nasloviti kot psevdospominsko lokacijo (v tem primeru sta to 2 byta; štirje biti so nezasedeni). Če ga želimo nasloviti s 24-bitnim naslovnim vodilom, moramo prek raznih logičnih funkcij povezati 23 bitov (A0 uporabimo za izbiro med enim ali drugim bytom)! Naslavljanje nekaj takšnih vezij na opisani način bi zahtevalo izredno veliko logičnih vezij, zato ga srečamo le poredkoma. Za izbiro CS raje uporabljamo dekodерje (sliki 5. 5 in 5. 6).



Slika 5. 5 Blokovna shema dekodерja SN74F138

Dekoder s slike lahko združi tri naslovne bite. Vsaka od osmih (2^3) kombinacij bo postavila le enega od izhodov v stanje 0, vsi ostali pa bodo setirani. Čeprav smo z njihovo uporabo bistveno zmanjšali potrebno število logičnih elementov, je očitno, da bo "široko" naslovno vodilo še vedno zahtevalo veliko dekodерjev.

Sprostitutveni vhodi			Izbirni vhodi			Izhodi							
G1	G2A	G2B	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
x	1	x	x	x	x	1	1	1	1	1	1	1	1
x	x	1	x	x	x	1	1	1	1	1	1	1	1
0	x	x	x	x	x	1	1	1	1	1	1	1	1
1	0	0	0	0	0	0	1	1	1	1	1	1	1
1	0	0	0	0	1	1	0	1	1	1	1	1	1
1	0	0	0	1	0	1	1	0	1	1	1	1	1
1	0	0	0	1	1	1	1	1	0	1	1	1	1
1	0	0	1	0	0	1	1	1	1	0	1	1	1
1	0	0	1	0	1	1	1	1	1	1	0	1	1
1	0	0	1	1	0	1	1	1	1	1	1	0	1
1	0	0	1	1	1	1	1	1	1	1	1	1	0

Slika 5. 6: Izjavnostna tabela dekoderja SN74F138

Opisani problem je možno elegantno rešiti s pomočjo PAL vezij (angl. Programmable Array Logic). To so čipi, v katerih lahko “programiramo” tudi zelo zapletene logične povezave vhodov in izhodov s postopkom, ki je podoben programiranju EPROM pomnilnikov.

5.5.2 Izbira čipov v MC68332

Pojektanti mikrokrmilnikov serije MC68300 so se zavedali opisanega problema. V ta namen so vpeljali programsko definirane CS izhode.

V ustreznih registrih SIM modula lahko določimo naslove za aktiviranje ene izmed dvanajstih mikrokrmilnikovih CS nožic (CSBOOT ter CS10 - CS0), ki bo povezana z vhodom CS perifernega čipa.

CSBOOT je rezervirana za t.i. *začetno nalaganje* (tudi začetni zagon, angl. bootstrap). V našem primeru pomeni to nalaganje inicializacijskega programa iz ROM (običajno EPROM) pomnilnika ob zagonu oz. resetiranju (reset prekinitvi ob aktiviranju RESET nožice). Zato mora imeti vsak dobro zasnovan sistem z MC68332 predviden že programirani pomnilnik, katerega CS je vezan na CSBOOT.

Preostale nožice (CS10 - CS0) so načeloma na voljo uporabniku.

OPOMBA!

V BCC izvedbi mikroračunalniškega sistema s PFB (pogl. 8) so vse CS nožice, **razen** CS3 pri “B” verziji, že uporabljene za naslavljanje določenih čipov (npr. CPU32Bug EPROM, RAM, eventualnega matematičnega koprocссора...).

Bistvo uporabe CS nožic v MC68332 je v možnosti softverske izbire naslova za njihovo aktiviranje. Za to sta zadolžena dva registra (slika 5. 7): CSBARx in CSORx (x ustreza zaporedni številki CS; glej sliko 5. 2).

Pri definiranju pomena posameznih bitov v registrih vseh podmodulov (SIM, QSM in TPU) veljajo naslednja pravila:

- prva vrstica vsebuje formalno ime⁵ in heksadecimalni naslov registra (v RAM),
- druga vrstica vsebuje zaporedne številke posameznih bitov,
- tretja vrstica vsebuje formalna imena posameznih bitov ali skupin bitov,
- četrta vrstica vsebuje stanje bita po resetiranju MC68332.

CSORx

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MODE	BYTE	R/W	STRB	DSACK				SPACE				IPL			AVEC
RES:0															
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CSBARx

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	BLKSZ		
RES:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Slika 5. 7: Registra CSORx in CSBARx

Pomen nekaterih polj v CSORx registrih:

BYTE:

- 00 - CS razveljavljen
- 01 - spodnji byte
- 10 - zgornji byte
- 11 - oba byta

R/W:

- 00 - rezervirano
- 01 - beri (read)
- 10 - piši (write)
- 11 - oboje

DSACK:

0000 - 0 čakalnih stanj (wait states)

.....

1101 - 13 čakalnih stanj

BYTE področje uporabljamo pri naslavljanju 16-bitne periferije. V tem primeru moramo določiti, ali naslavlja CS spodnji ali zgornji byte besede ali pa oba (glej primer na sliki 5. 9).

⁵ Mnemonik registra (v tem primeru CSORx) in mnemoniki posameznih bitov ali skupin bitov (tukaj MODE, BYTE itd.) nimajo nobenega konkretnega pomena. Njihov namen je le opozarjanje na funkcijo, ki jim je namenjena. V programu moramo tem simbolom dodeliti dejanske naslove (npr. register CSOR3 naslavljam s tem simbolom le, če smo mu dodelili naslov Yffa5a_{HEX}, MODE bit je le 15. bit tega registra itn. Glej primer programiranja v pogl. 7.2).

Pri naslavljanju je LSB naslovnega vodila (ADR0) zadolžen za ločitev med spodnjim in zgornjim oz. sodim in lihim bytom.

Naslavljanje enot, ki dovoljujejo le branje (npr. ROM pomnilnikov, A/D pretvornikov itd.), določimo v polju **R/W** z bitoma 01. Kombinacija 10 aktivira **CS** le pri vpisu v enoto (npr. D/A pretvornik), če pa se bo signal za izbiro čipa aktiviral pri obeh operacijah (npr. ob naslavljanju RAM pomnilnikov), postavimo vsebino polja na 11.

Prilagajanje delovanja MC68332 počasnejši periferiji lahko dosežemo z uvajanjem čakalnih stanj. V polju **DSACK** definiramo njihovo trajanje z mnogokratnikom periode sistemske ure.

CSBARx register določa naslov in dolžino pomnilniškega bloka, ki ga želimo naslavljati (slika 5. 8).

Polje BLKSIZE	Dolžina bloka	Upoštevani biti naslovnega vodila
000	2 K	A23 - A11
001	8 K	A23 - A13
010	16 K	A23 - A15
011	64 K	A23 - A16
100	128 K	A23 - A17
101	256 K	A23 - A18
110	512 K	A23 - A19
111	1 M	A23 - A20

Slika 5. 8: Kode za register CSBARx

Biti 15 - 3 določajo naslov (biti na naslovnem vodilu A23 - A11), na katerem se nahaja spominski blok dolžine, ki je določena s poljem BLKS (angl. Block Size). Če želimo na naslov F00000_{HEX} postaviti 8 Kbytni pomnilnik s slike 5. 3, moramo BLKS maskirati v stanje 001. Naslov bloka pomnilnika vpišemo v naslovno polje. Iz tabele je razvidno, da bodo pri definiranju tega naslova sodelovali le biti A23 - A13, kar smo ugotovili tudi v primeru iz podpoglavja 5.5.1. Pri tem bita A11 in A12 nimata nobenega vpliva, saj ju, skupaj s biti A0 - A10 rabimo za določitev interne lokacije v RAM.

Kombinacija bitov za omenjeni primer je:

$$(CSBAR) = 1111\ 0000\ 000x\ x001_{BIN}.$$

Pri naslavljanju katerekoli bytne lokacije med F00000_{HEX} - F01FFF_{HEX} se bo **CS**, ki smo ga določili za ta RAM, avtomatično postavil v aktivno stanje (0). Na ta način smo se izognili potrebi po hardverskem določanju **CS**.

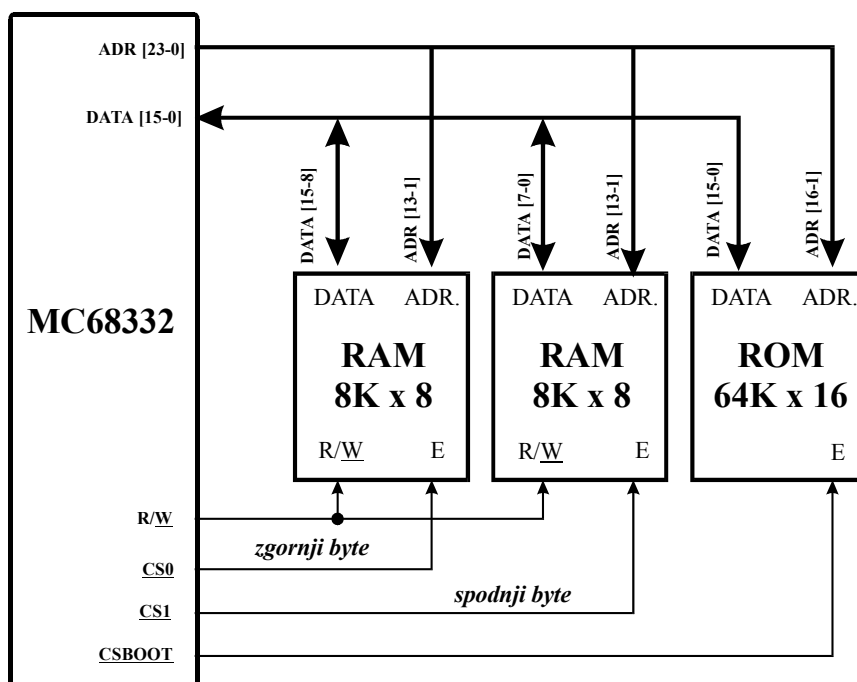
5.5.2.1 Primer naslavljanja pomnilnikov

Tipičen primer naslavljanja pomnilnikov ob uporabi CS izhoda ter vodil kaže slika 5. 9 [24]. V tem primeru imamo opravlka z dvema RAM pomnilnikoma velikosti 8 Kbyte (8 K x 8 bit) in enim “boot” ROM (ali EPROM) pomnilnikom velikosti 64 Kword (64 K x 16 bit).

Podatkovno vodilo ROM je povezano z ustreznimi biti zunanjega 16-bitnega podatkovnega vodila MC68332. Podatkovno vodilo prvega RAM je povezano z MSByte podatkovnega vodila mikrokrmilnika, katerega spodnja polovica (LSByte) pa je povezana z drugim RAM pomnilnikom. Naslovni biti MC68332 (A13-A1) so povezani z A12 - A0 pomnilnikov, torej so premaknjeni za en bit v levo!

Razlog za takšne povezave je enostaven. Mikrokrmilnik bo iz perifernih pomnilnikov bral podatke dolžine ene besede (16 bitov). Pri vsaki besedi, ki jo preberamo iz RAMa, se bo ena polovica nahajala v enem, druga pa v drugem čipu. Torej smo s tako konfiguracijo dosegli ekvivalent RAM pomnilnika velikosti 8 Kword. Z ozirom na hkratno naslavljanje obeh RAM pomnilnikov⁶ moramo tudi CS0 in CS1 prožiti hkrati. Zaradi naslavljanja bytov z različno težo moramo BYTE področje CSOR1 registra postaviti v stanje 01_{BIN} (CS1 naslavlja spodnji byte), CSOR0 pa v 10_{BIN} (CS0 naslavlja zgornji byte)⁷.

Naj bo vsebina CSBAR0 in CSBAR1 enaka 1001_{HEX}. S tem smo za RAM rezervirali področje 2 x 8 Kbytov, ki se začne na naslovu 100000_{HEX}.



Slika 5. 9: Primer naslavljanja zunanjih pomnilnikov

ROM pomnilnik vsebuje celice dolžine ene besede, kar pomeni, da posameznih bytov ne moremo nasloviti. Zato so tudi tukaj biti naslovnega mikrokrmilnika ADR16-1 povezani z biti

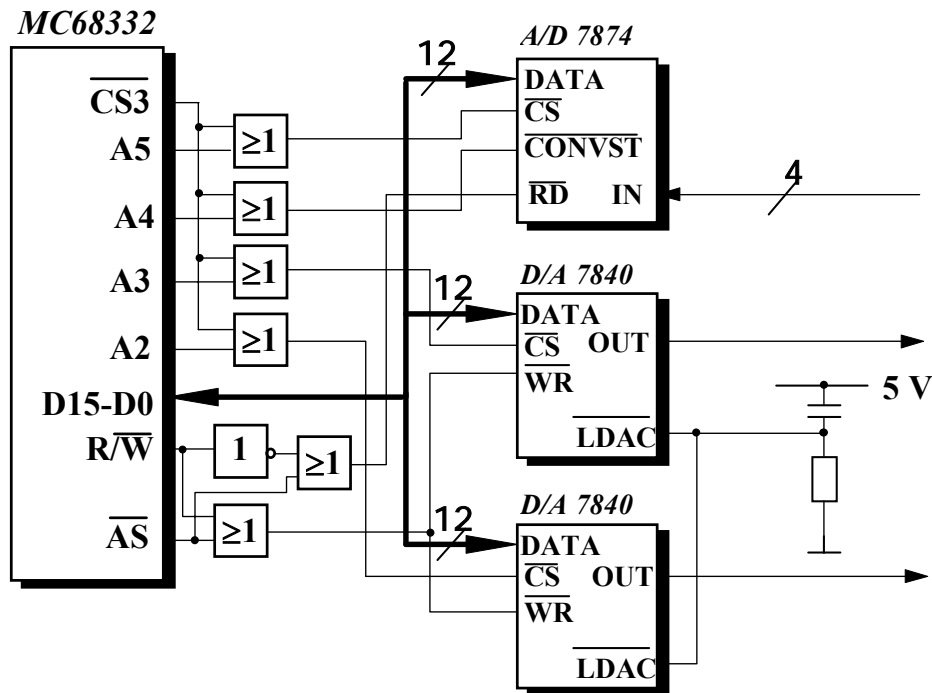
⁶ PAZI! V tem primeru hkratno naslavljanje ne pomeni tudi konflikta na podatkovnem vodilu, saj pošilja vsak čip hkrati podatke po svoji polovici vodila.

⁷ CS signali mikrokontrolerja so povezani z vhodi E (angl. enable - omogočiti) na čipih (glej tudi pogl. 12.1).

ADR15-0 pomnilnika. (CSBARBT) = 0003_{HEX}, kar določa aktiviranje CSBOOT za naslavljanje področja velikosti 128 Kbyte = 64 Kword na začetnem naslovu 0.

5.5.2.2 Primer naslavljanja majhnih enot

Najmanjše področje, ki ga na ta način naslovimo, je 2 K (slika 5. 8). Če pa hočemo nasloviti manjše področje (npr. A/D ali D/A pretvornik), moramo preostale bite (A10 - A0) ustrezno dekodirati. Slika 5. 10 kaže možno rešitev izbire dvobytnih lokacij brez potrebe po dekodiranju.



Slika 5. 10: Uporaba CS3 pri naslavljanju dvobytnih lokacij

V tem primeru naslavljam tri čipe: 12-bitni A/D pretvornik in dva 12-bitna D/A pretvornika (pogl. 12.4). Pri izbiri čipov bo sodeloval CS3. V fazi inicializacijskega dela programa bomo registra za konfiguriranje CS postavili v naslednje stanje:

$$(\text{CSBAR3}) = 0500_{\text{HEX}} = 0000\ 0101\ 0000\ 0000_{\text{BIN}}$$

$$(\text{CSOR3}) = 7b30_{\text{HEX}} = 0111\ 1011\ 0011\ 0000_{\text{BIN}}$$

S prvim registrom smo določili osnovni naslov 050000_{HEX} najmanjšega spominskega bloka (2 K). Drugi register pa definira način naslavljanja: asinhronski prenos, možnost posamičnega naslavljanja sodega in lihega byta, proženje CS pri branju (read) in vpisovanju (write) ob veljavnem naslovu na vodilu (AS), 12 čakalnih stanj (wait states) itd. Znotraj bloka 2 K sodelujejo pri naslavljanju naslovni biti A10-A0 (slika 5. 11).

2. nibble			1. nibble				0. nibble				Izbrani čip
A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	
X	X	X	X	X	0	1	1	1	X ⁸	Y ⁹	CS AD7874
X	X	X	X	X	1	0	1	1	X	Y	CONVST
X	X	X	X	X	1	1	0	1	X	Y	CS 1. DA7840
X	X	X	X	X	1	1	1	0	X	Y	CS 2. DA7840

Slika 5. 11: Kombinacije bitov A10-A0 za izbiro posameznih čipov s slike 5. 10

Izbrana arhitektura s slike 5. 10 namesto zapletenega dekodiranja enajstih bitov uporablja le štiri bite A5-A2. Posamezni čipi bodo naslovljeni le, če je **eden** izmed teh bitov 0, **vs**i ostali pa 1. Biti A10-A6 in A1 lahko imajo **poljubna stanja**. Izjema je le bit A0, ki je 0, če hočemo nasloviti spodnjih osem bitov pretvornika oz. 1, če naslavljamo zgornje štiri bite. To dejansko pomeni, da npr. A/D pretvornik naslovimo na več načinov (05001e_{HEX}, 0500dc_{HEX}, 05001d_{HEX}, 05005d_{HEX} itd.). Pomembna je le kombinacija 0111 na nožicah A5-A2!

Čeprav je zgornji način nekoliko nekonvencionalen, izpolnjuje osnovno zahtevo, ki pravi, da z enim naslovom ne smemo nasloviti več čipov. V takšnih arhitekturah lahko zgornjo idejo uporabimo ob dodatnem sodelovanju dekodejev. Osnovna pomanjkljivost opisanega pristopa je v tem, da smo področje 2 Kbytov spominske mape MC68332 uporabili za naslavljanje le 2 x 4 = 8 bytov.

⁸ X je poljubna vrednost: 1 ali 0.

⁹ Y je 0, če naslavljamo sodi bajt (spodnjih osem bitov pretvornika) in 1, če naslavljamo lihi bajt (zajeti so štirje zgornji biti pretvornika).