МПСхТ - Микропроцесорна Схемотехника

проф Р. Иванов — лаб. упр - 1 седм

изпит - 2 въпроса

① Вградени микропроцесорни с-ми (Embedded system)

MPU - микропроцесор

MPC - микроконтролер

DSP - цифрови процеси

Това са системи за ориентирано управление на даден обект.
Има два вида с-ми свързани с обекта - аналогови и цифрови.

Блокова схема: Фон-Нойманова архитектура

формират се 3 магистрали:
- адресна - излиза от МРС и отива към другите блокове
- магистрала за данни - двупосочна т-е в един момент процесора може да записва данни, а в друг да чете данни. Определя разредността на микропроцесора - т-е колко бита паралелно ще се обработват.
- управляваща магистрала - включва набор от с-ли дава възможности да синхрониз. работата на отделните елементи
Това позволява да имаме по-проста архитектура.
Тази архитектура е по-бавна в сравнение с Харвардската архитектура.

RAM - памет, която може да се чете и записва произволно, използва се за запис на данни (~100 Mbite) до няколко десетки kbite). Тази памет помни докато има захранване.

<u>Прогр. памет</u> - тук се записват инструкциите на програмата. Тя е постоянна. Може да бъде три типа:
- ROM памет
- EPROM памет - с ултравиол. светл. се изтрива - представлява плаващ гейт, в който при внасяне на електрони става запис, но при специален режим

- керамичен корпус с кварцово прозорче по-скъп в сравнение с (ОТР)

ОТР - с пластмасов корпус и няма прозорче

- Flash памет - по електрически път се записва и изтрива. Поставя се в пластмасов корпус; Имат вграден умножител за напрежение, така че лесно става препрограмирането на паметта
Най-масово използваната памет

I/O - (входно-изходно у-во) - интерфейс - осъществява връзка с обекта за управл.
Два режима - четене и запис
- режим запис / четене

МРУ - представлява изхода блокова схема взета заедно включва МРУ (8, 16 bite), RAM паметта (движи се от 128 байта до 2x bait), програмна памет (1xB ÷ 60xB). Вх/изх у-во, което е най-важното за МСУ. Вх/изх у-во (интерфейс) бива изколко вида: - паралелен, сериен, таймер, АДС и др

8 битов МРУ обработва 255 числа

16 битов МРУ обработва до (64к) числа ⇒

⇒ две паралелни памети са необходими

Развитие на пазара:

4 битовите почти изчезват заместват се от 8 битови
DSP навлизат повече в приложенията

DSP - цифрови памети - различава се по това, че архит. е Харвардска. Но - висока производителност (измерва се в мипсове - MIPS). Широко приложение в комуникациите. Производителността в DSP се плаща с консумация. Напр: DSP от вида TMS 320LF2406 има консумация 100 mA, а 16/32 битов → от вида MSP430F149 консумира 0,35 mA. DSP работи на 40 MHz, а 16/32 на 1 MHz.

Структурна схема на МРУ (2)

АЛУ - аритметико - логич. у-во - за извършване на аритметични операции

CCR - (регистър на условията) - дава инф. след извършване на аритм. опер. с АЛУ кой флаг се вдига

ACC - акумулатор - разредността му е колкото на процесора. Брой им е различен.

Прекъсвания - маскируеми и немаскируеми. Позволяват многозадачен режим.

Модул за управление - изпълнява отделните инструкции

Блок с регистри:

- PC - програмен брояч - записан е адреса на следващата операция, която ще се изпълнява

- SP - указател на стека - показва къде се намира стековата памет. Тя е необходима за да се изпълняват прекъсвания
- IX - индексни регистри - за индексна адресация при адресиране и тяхното съдържание да се изведе като адрес навън.

Такава структура има напр. процесор 58HC11

2. Видове инструкции

Програмен модел на микропроцесор - програмно достъпни регистри

Програмен модел на MC68HC11A1 - включва няколко регистъра акумулатори A и B - това са регистри с общо предназначение с тях можем да записваме и да четем. Те могат да бъдат групирани в D регистър - 16 битов, то A регистър е старшата част, а B регистър е младшата D и A и B е един и същ регистър

A	8	B	8
D			16

IX	16
IY	16
SP	16
PC	16

CCR	S	X	H	I	N	Z	V	C

Индексен регистър X и Y, чрез който можем да правим т.н индексна адресация

- указател на стека SP - т.н стекова памет

Програмния брояч PC - регистър, който съдържа следващата инструкция която ще се изпълнява

CCR - регистър на условните преходи - в него се вдигат един или друг флаг

... C - използва се не само за събиране, а и за проверки

V - препълване, свързан е с най-старшия бит

Z - ако резул. на действ. на АЛУ е получава 0 този флаг се вдига

N- отриц. число с резултата и тогава този флаг се вдига
I- флаг маска за прекъсване, свързан с сис. забрана и разрешаване на прекъсванията
H- полупренос
X- свързано е също с прекъсвания, но с друг вход
S- стоп, когато процесора може да се спре
HC11 има 21 вида прекъсвания

<u>Видове инструкции</u> - зависят много от микропроцес. (MPU)

V инструкция започва с код на инструкцията (КИ)
1. байтовите инстр се състоят само от КИ
двубайтовите I инстр е КИ, а II е данни или адрес

3-байтови - I байт е КИ, а следващия може да бъде данни или адрес

PI

14 - битова дума

MSP430

<u>Изпълнение на инструкция</u> - бр. за изпълнението на разл. инстр. е различно от 2÷6 такта, тъй като работи на 2MHz; 0,5 μs т.е за изпълнението на 1 инстр. е необходимо 1÷3 μs

Адресно поле на една такава програма - за да стартираме програмата в програмния брояч трябва да е 1000. Винаги програмата трябва да започва с КИ,

защото МРУ чете кода на инструкцията и разпознава

дали тази инструкция е едно байтова, 2- и 3 - байтова. По този код разпознава, че следващия байт е данни. КИ и данните се заемират на едно и също място в едно и също поле.

Видове адресации

1. Акумулаторна - TAB - (трансфер акумулатор А в акумулатор В) взема съдържанието от акумулатор А и го записва в акумулатор В. При него ни интересува само КИ.
2. Вътрешна (подразбираща се) - свързани са с действия на вътр. регистри
3. Непосредствена

LDAA # $0F 86 0F

- състои се от 2 байта - първия е КИ, а вторие е данни

$ - шестнайсетично

- знак за непосредствена адресация

Непоср. адресация ни дава възможност в акумулатора да зареждаме различни числа

4. Директна адресация - тя е двубайтова и дава възможности за запис и четене на първите 0÷255 байта от адресното поле

LDAA MEM

Някъде сме декларирали че MEM е = 20 => ще прочете съдържанието на клетката MEM и ще записва в паметта

96
20

- КИ

MEM - адрес от паметта

- по-бърза и по-къса инстр

5. Разширена адресация - за адресиране на цялото адресно поле

LDAA MEM

B6
0010
20

0020
1000

При нея освен КМ имаме 2 байта за адрес

Тя покрива директната адресация, но е по-бавна, тъй като покрива 3 байта

6. Индексна адресация - използва се индексния регистър

LDAA 0,X

A6
00

В него предварително записваме някакво число

LDX # $1020

Предполага се, че преди изпълнението на инструкцията сме записали в индекс регистър адреса от който ще четем. Тя позволява да правим едно отместване 0,X ⇒ тази инструкция чете клетка от паметта записана точно в индексния регистър

20,X - ще прочете същия адрес

Тя е много удобна при четене на масиви

7. Относителна адресация

в границите → -125 до +128

От една инстр. може да се отмества една инстр. в границите -125 до +128

BRZ L1.

Свързана е с проверката на отделни флагове

<u>Видове инструкции</u>

- Аритметични - за събиране и изваждане, умножение и деление, но те са ограничени - 16 битови, целочислено
- Управляващи - с управление на регистри свързани с прехвърлянията и тяхното управление

- Побитови - инструкции, които дават възможност MPU побитово да работим
- логически инстр.

Интерфейси при микропроцесорните с-ми (3)

Чрез интерф MCU се свързва с обекта, който ще управлява, с параметр. които ще измерва

Тези интерфейси можем да ги разделим на:
- Паралелни
- Серийни

Паралелни - Ако MPU е 8 битов - основното предимство е че тк са бързи и можем да прочетем цялото число

Паралелните:
- Входни ⇐
- Изходни ⇒

} циф вх и изх

- Двупосочни ⇔

изходни стъпала

1)

1) Пред: - само с един транзистор се реализира

Изх стъпала са много важни

Не е без значение какво ни е крайното стъпало

- отворен колектор
- пускови стъпала

Входни стъпала

CMOS входовете се свързват към маса или към +5 или нула, защото от статично електричество могат да се отпушат и да протече ток. Поради това при микроконтролерите се слага допълнителна схема

Схема, която не позволява междинно състояние - ще бъде или 0 или 1 на входа

Съпр. вх. съпротивление дава възможност тази схема да се използва като вх

Двупосочни - комбинация от вх. и изх

за HC 11

Port C е двупосочен
Port B е изход
Port E е вход
Port-овете са 8 битови

Регистри за управление на port

- вх. регистър - от него четем данните т-е porta
- изх. регистър - можем да пишем в porta
- DDR - регистър в който ∀ бит показва съотв. извод дали е вход или изход
- Int flags - флагове за прекъсвания

Има 3 регистъра свързани с прекъсванията
Може да изберем по ~~някакъв~~ какъв фронт ще се извършва прекъсването - по спадащ или нарастващ

- функцион. регистър

Управлението на един 8 битов port може да стане до 7 регистъра, може да е и един.

Пример: CM602

при MCU
Организацията е по 8 бита
8, 16, 32
HC 11 – портове A, B, C, D, E, F, G
P_0, P_1, P_2, P_3

MC 68HC 11 A_1
Корпус 40 изводен – порт E – 4 битов
Корпус 44 изводен – порт E – 8 битов

Серийни интерфейси (4)
– асинхронни
– синхронни

Пример:

Серийния интерфейс е регистър, в който записваме данни паралелно

Регистър в който можем да запишем данните паралелно и след това да ги преобразуваме в поредов.

Асинхр – предава се 1 байт предават се само данните
Скор. на предавателя и приемника трябва да е една и съща
Предполага включване на стартов бит – 0; 2 стопови
Освен данните не се предава тактова честота
∀ бит се предава за едно и също време

– за проверка 01010101 (55)
01010100 (54)

7 бита + четност
при четност – бр. „1" е

→ по 16 такта се предава 1 бит ⇒ честотата расте 16пъти за да намалим вероятността за грешка

13

Синхронен – освен данните се предава и тактовата честота. Имаме водещ Master и Slave

1. Асинхронни – ACIA интерфейси

$f_{пр}$ – подаваме честота $f_{пр}$

RxD – приемника

регистри – 4бр

TxD само за запис

RxD само за четене

предаване на 8 бита данни + 2 стопови

CR | 7 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | – Управляващ регистър (контролен)

IRQ; RS (RTS); формат: 00 – 1; 01 – 16; 10 – 64; 11 – Reset

Обикновено TxCl и RxCl са свързани на кръст, което означава, че честотата на предаване е една и съща, но може да бъде и различна.

SR | 7 | | | | | CTS | DCD | TxD | RxD | 0 | – регистър на състоянието (8 битов)

формат на грешка

прекъсване

четност (автоматичност)

IRQ – заявка за прекъсване

RxD – показва и се вдига когато регистъра е пълен (приемащ регистър)

TxD – показва дали записаната инф. е предадена (предаващ регистър)

зададения формат и получения се различават – формат на грешка

Примерни програми за предаване и приемане:

Винаги програмата трябва да прави инициализация

Инициализация за дадената схема:

непосред. адрес. – LDAA #03 – коефициент на деление 16

STAA ACIACR – адреса на контролен регистър

LDAA #$11 – в акумулатора A се записва

STAA ACIACR

Програма за четене и запис/запис

L1 LDAA ACIA5R – в акумулатора A се записва сост. на статус регистъра

LSRA
LSRA – съдържанието на най-младшия бит отива в пренос
BCC L1

LDAA ACIAD
STAB ACIAD

Недост: процесора се върти в цикъл и ако нищо не дойде, настъпва безкраен цикъл

B – съдържа числото, което искаме да предадем

2. Синхронни интерфейси (SPI)

Разлики м/у асинхронни и синхр. – при синхр. интерф. не се предават допълн. битове – само данни и тактова честота

Предават се само данни и се тактува с честота

Разглеждаме две устройства при синхр. интерфейси:

* Мастер – дава тактовата честота
* Slave

Друг синхр. интерфейс (I^2C) – имат добавени елементи

когато такта L и фронта пада се стартира в обратния случай се стопира

Приложение на SPI в интегрални схеми:

Синхронните интерфейси са много употребявани при вградените микропроцесорни системи. Синхронните интерфейси дават възможност за разширяване на системите като позволява изнасяне на някои модули извън системата

Времеви интервали. Генериране и измерване (5)

– софтуерно

	инструкция, която вдига 1 бит в 1	
	LDAA #$FF	
L1	DECA	– намаляване на съдър. на A
	BNE L1	

– хардуерно

Таймерните системи включват: брояч – 8 – 16 битови и по-рядко 32, делители – за да намалим вх. честота, блокове за измерване на времеви интервали, блокове за генериране на времеви интервали, COP-системи, схеми за начално reset установяване, осцилатори, таймерни системи свързани с Real time-тактуване, часовници за реално време

Съвременните броячи работят с натрупване и изваждане

Измерване на временни интервали:

Брояча непрекъснато брои. Микропроцесора може да обслужва модула по два начина:
- чрез flag
- чрез INT

X_1 - число

X_2 - число $\neq X_1$

$\Delta x = X_2 - X_1$ - число което зависи от брояча

през T, брояча си е увеличил съдържанието с Δx, като си умножи с T се получава време.

0,5μs, ако $\Delta X = 4$ 2μs ± 1 дискрет

Ако работим на ниска честота

MC68HC11 осцилатор → 8MHz → таймер 2MHz

PIC 16F84 осцилатор → 20MHz → 5MHz (200ns)

Генериране на временни интервали:

ц.-к. - цифров компаратор, сравнява състоянието на брояча и регистъра, когато числото на брояча си е равни с числото на регистъра се получава строб.

MPU записва X_1 в регистъра, когато брояча стигне X_1 си получава строб

записваме $X_2 = X_1 + \Delta X$, брояча продължа да брои, и се получава друг строб

Използване на ШИМ - Широчинно импулсна модулация

Таймерна система за следене и изпълнение на програмата
- Watch dog
- COP (при Motorola)

ЧМ - чакащ мултивибратор
МСО - микроконтролер

ЧМ - изработва импулс с продължителност 1 ms и ако не подадем определено време импулс то ЧМ подава на Reset
→ това е така наречения Watchdog
COP системата не се включва в началото

ШИМ ОС

измерв. на вр. IC

WD

таймер
брояч

16 битовия брояч, който се подава на 4 отделни модули
ОС - модули, които генерират импулси
Правят се няколко такива модула, за да се управляват няколко изхода. При HC11 са 5 такива модули
Ic - при HC11 са 3
IC - Input Capture
Таймерните системи са съществена част от периферията.
Един брояч се използва за няколко модула

Таймерни системи при DSP

Различават се от таймерните системи при микроконтролерите:

- работят на по-висока честота
- имат от 2÷3 броча
- Управление за ADC – част от тайм. с-ма позволява АЦП синхр да се пуска с тази с-ма
- най-често над 6÷8 mix PWM
- има State mashine – за векторно управление на мотори (хардуерен модул)

- модул dead band – имаме 2 ключа, които имат вр. закъснение и управл. на двата ключа да са дефазирани и този модул е да си прави това закъснение
- QEP – логика, която позволява да се четат оптични N-кодери

един оборот 500÷1000

С N-кодера се мерят оборотите. Тези N-кодери дават на 1 оборот 1000 импулса т.е имат висока точност ?

Часовници за реално време (ЧР)

Таймери с особено предназначение. Таймер който непрекъснато трябва да работи

Интерфейса може да бъде паралелен
Синхронен интерфейс – SPI; I²C
Има кварцов резонатор – най-често на 32 kHz, тя е CMOS схема

Най-често първите 32 байта е RAM памет
има регистри за sec; min, часове, дни от седм, месец и год.
има и управл. регистри + аларми. Всички тези регистри са 8 битови. Изход с-тактова честота
Основно е ЧР да имат малка консумация

Схеми за начално установяване

Захранването не може да скочи веднага на 5V, има опред. закъсн. С настъпване (нарастване) на захр. в един момент стартира инер. и в брояча се появява произволно число. Понякога при отпадане на захр може да се получи мин V и часто падне V не се знае как ще се държи тази схема. Той до изкеде пак ще функционира. И при такива моменти трябва нашата сма да се ресектира

Пример за една такава схема

TL7705A

K_1 и K_2 - компаратори

Reset in - вход

sens - чувствителност

На входа на K_1 има генер на ток 100μA, който отива на вх. С. Към другия вх е вкл. вътрешен тиристор, който се управлява. K_2 чрез делител може да следи вх. U изх. на K_2 отива към управлението на тиристора
Sense входа се свързва на 5V. В момента, когато U падне, K_2 ще сработи и ще отпушчи тиристора.
Ако подадем захр. външ. кондензатор не е зареден и ще започне да се зарежда с ток 100μA и U му ще нараства и U в т. (вх на K_1) ще стане по-голямо от Reset и K_1 ще сработи. Ако се получи $V^{пик}$, ще влезе през sense и K_2 ще сработи и ще пусне тиристора и ще разреди кондензатора за кратко.

Генератори

Усилвателното стъпало трябва да отговаря на чет. лента. Фешка. Не може едно и свито усилвателно стъпало да се вкл. към 32 kHz и 10 MHz.

Затова външно трябва да се вкл. С. Важно е генер. да е стабилен и затова е поставен през RC група и MOS транз.

Унф осцилатор с RC група.
По унф. път може да се препрограмира. Със вкл. на захр. почва да работи
Недост: на тази с-ма е че не е стабилна (и t^0)
Има 2 осцилатора, първия е стъпало, което може да работи

и на 32 kHz и 1÷8 MHz, II осцилатор може да работи само от 1÷8 MHz. Тей като този микроконтролер е да работи на малка консумация

Видове памети. Организация на паметта (6)

27C16 (8)
27C64 (8)
27C256 (8)

Дефинират се два вида памети:

1. RAM - при изкл. на захр. инф. изчезва, тя е за временно съхраняване на паметта (статична памет)
2. - Програмна памет бива:

ROM - прави се от производителя на интегралната схема тя е по-евтина, но недост е че ни трябва 10000

EPROM - с ултравиолетово изтриване. Организирана е на базата на плаващи гейтове и има заряд. Целта е да избием този заряд. Записа в тази памет е да запишем нули да вкараме заряд в плав. гейт.

Тя се програмира на спец. програматор

Flash - (съвременни) - електрически изтриване и електр. програмиране. Директно на самото у-во могат да бъдат препрограмируеми

RAM - тя е статична памет

Организация на паметта е напълно достатъчна

RAM - 2k, 8k ; 32k

6116(8) 6264(8) 62256(8)

OE – управлява изх. буфери при четене

С-ите идват директно на едни и същи изводи

Ако процесора ни е 16 битов, паметта се организира така:

младшия се записва в едната памет, а старшия в другата

– Паралелно свързване

8к x 16

Ако искаме да свържем две памети последователно:

CS са разделени и в единия случай подава $\overline{CS_1}$ и тази памет отговаря и така можем да си направим организация
8k $\overset{?}{\times}$ 16k
16k × 8 bit

<u>EEPROM</u> - използва се за съхранение на някакъв резултат (фонокарти)

За запомняне на константи, възможно е и тези константи да ги променяме. Записа и четенето е последователно - бавно е

Осн. се използва за разл. коеф, които трябва да се променят след известно време

<u>Flash</u>

Am 29F010

131 k × 8 бита

Обслужването за блокова схема от "Видове памети" бива:
- Програмно
- с прекъсване

Имаме два вида режима - запис и четене

Обслужването се извършва програм. вх-изх у-во - трябва ∀ събития и процеси да се следят последователно

(7) Работа с прекъсвания (програмно) – тук проверката е изнесена (обслуж. на прек.)

- Програмно
- Работа с прекъсване

Работата с прекъсвания довежда до оптимално използване на програмата

Проблеми: Ако заявката е от 1 източник – проблеми няма

Когато имаме повече от една организация за прекъсване трябва да имаме регистри и да се разпознае от къде идва това прекъсване. Ако заявките се подават едновременно тогава се използва приоритет за даден флаг

Система за обслужване на прекъсванията

– дава възможност за разрешаване или забраняване на прекъсването

– регистър за приоритет – показва кое събитие да се изпълни първо

Прекъсванията позволяват програмата да се разтоварва

При HC11 прекъсванията са 21 вида

Има и ДМА – (директен достъп до памет) – той се налага – ако искаме да четем данни от вън.

Микропр. и оброца към дадено у-во чете данните, записва ги

в акумулатор, а неговото съдържание в RAM паметта.
На адресната магистрала се поставя ДМА контролер – чрез него може да се чете от външното у-во и да се записва във вътрешната памет на микропроцесорните с-ми за 1 такт.
При директен достъп до паметта още при проектирането се вгражда такъв ДМА контролер.

(8) Адресно декодиране на микропроцесорните с-ми

Адресна магистрала на HC11 – разгл. адресно поле от 64KB

За HC11 адресната магистрала е 16 бита

Прави се адресно декодиране:

Взима се старшата част на адреса и се подава на дешифратор: 74HC138

Ако използваме такъв дешифратор разделяме адр. поле на 8 зони по 8KB

Пълно декод. – всички адресни линии участват в декодирането
Нетълно декодиране – не участват 4 адресни линии

Y_3 – чип селект на дешифратора
от A_2 до A_{12} – са свободни

От какви съображения се избира кое къде да се постави
- RAM – може да се постави навсякъде
- EPROM → Flash
- I/O / PIA

Векторите за прекъсване се поставят в постоянната памет

Паметта се поставя там където стоят Reset векторите, ако EPROM се постави

RAM – поставя се в нулевата зона

другаде, трябва да се осигури още една памет за Reset векторите

Несиметрично декодиране – 1 дешифратора с 4 изхода

С първия дешифратор разделяме адресното поле на 4 части по 16 kB.

Дешифрация с цифров компаратор – 4 битов

Дешифрация се прави и с TTL PROM – памет с адреси, програм. се еднократно

Дава възможност за промяна на адресното поле →

→ Таблица на истинност

A_2	A_1	A_0	D	
0	0	0	11111110	FE
0	0	1	11111101	FD
0	1	0	11111011	FB
0	1	1	11110111	F7
1	0	0	11101111	EF
1	0	1	11011111	DF
1	1	0	10111111	BF
1	1	1	01111111	7F

Декодиране на микроконтролерите – при микроконтролерите всичко е поставено в една интегрална схема. Дават възможност за адресиране. Имат по няколко CS. Някои имат паралелни модули

(9) Основни етапи на проектиране на електронно-изм. у-ва

1. Техническо задание
2. Разработка
3. Реализация на разработката
4. Внедряване

Блокова схема на процеса на разработка

Хардуера и софтуера могат да се проектират отделно. Проектират се на асемблер използва се малък код с голямо бързодействие а общо проектирането може да се извърши на Си. Има симулатори за тестване на софтуера.

Основ. разлика на тези системите е че трябва да се направи настройка на хардуера и софтуера съвместно, от където произтичат проблеми. Тази съвместна настройка може да се реализира с вътрешно схемни емулатори – микроп. системи на базата на микропроцесор, др. начин е чрез ROM емулатори, или чрез ICD – които позволяват директно да се записва в Flash паметта.

Организация на микропроцесорни с-ми

Блокова схема на микропроц. с-ма

Организацията на микропроцесорна система бива:

1 – модулна
2 – едноплаткова
3 – смесена

1 <u>Модулната</u> организация е исторически първата, която е използвана, облекчава системата чрез използването на микропр. модул, който се използва за всичко. За да се изгради такава модулна организация, никой не може да каже колко модула ще има.

Проектирането на 4 модул се извършва с буфериране на сигналите, увеличава с-мата, и повишава консумацията. Позволява много бързо да се конфигурират различни сми. Позволява бързо разработване на нови изделия, но на по-висока цена. Използва се при реализиране на малки серии, но за по-големи серии е икономически непрогодно.

2 <u>Едноплаткова</u> – предимства: тъй като се проектира специално за конкретни цели не се налага буфериране, слагаме толкова памет колкото е необходима. Имаме минимизиране на хардуера, намалява се консумацията.

Нед: ако трябва да променим нещо, това не може да го направим налага се изграждане на нова платка. Едноплатковите са предназначени за големи серии.

3 <u>Смесена</u> – на принципа на персоналния компютър. В единия край има жнлини (за дънна платка). Има възможност за разширение. Имали една дънна платка, която е базова и модули, на които да правим разл. конфигурации. Тя е комбинация от модулна и едноплаткова.

MC68HC11A1 – Един микроконтролер, който е изгрупо статичен работи от 0÷8 MHz. Той е изработен по HC-технология - едновременно е и микропроцесор и микроконтролер

Блокова схема

Включва 8 битово CPU в него има вграден ROM-8k, 512-EPROM (изтриваема памет) и RAM-256 има и вграден осцилатор и външно и вкл. резонатор, CPU работи на 2MHz, има модул таймер 16 битов. Port A – 8 битов директно свързан

Bus exp – разширение на магистралата. Зависи сам, Ч само изх, а и вх и изх. Намира се на адрес 1000

Port B – само изход

Port C – и вх и изх, двупосочен

Port E – само входове

Port D – една част са вх., а др. са изходи, ако използваме интерфейси губим Port D

Може да работи за няколко режима, които се определят от два входа (mode A и mode B)

mode A	mode B	
0	1	едночипов режим – не се ползва много
1	1	разширен режим – port B и port C се използват за адресна магистрала
0	0	– тестов едночипов
1	0	– тестов разширен

за данни port B и port C се превръщат в магистрали

тестов едночипен – процесора отива в D зоната, по серийния адрес чете 256 байта данни и се стартира
Тестови разширен – векторите не отиват към F зоната, а към B зоната. И е позволено в този режим и не се изисква да се работи в него

MC68HC11A1 – има два основни режима на работа едночипов и разширен

Едночипов режим

$\overline{IRQ}$ – заявка за прекъсване

$\overline{XIRQ}$

OSC – фактическата тактова честота е 2 MHz

Разширен режим

В разширен режим port B и C променят функциите си, а другите остават същите

Времедиаграми при работа в разширен режим

Данните се четат или записват по падащия фронт на E

– Няколко съществени регистъра от структурата на HC11 (от общо 64 регистъра)

1) CONFIG намира се на адрес 103F, той е EPROM памет, но в нея може да се записва при определени условия.

Променям регистъра и даваме Reset. Трябва да забраним вътрешния ром при работа с CONFIG

COP – може да се забрани достъпа

За да може да работи в разширен режим в CONFIG регистъра трябва да пише 0D

Адресно поле – за да видим разположението на отделните модули

2) INIT намира се на 103D – свързан е със мястото на RAM паметта

Позволява преместването на RAM паметта и рег. блок в адресното поле

Първите 64 такта е разрешена инициализацията, с цел защита

Как работи COP системата?

COP – дава възможност да се следи дали микропроц. изпълнява зададената от нас програма

COP системата се пренарежда, когато програмата е изцяло изчистена

Разглеждаме последователно портовете

port A се намира на адрес 1000 (8 битов порт)

IC – трите младши бита са сами за изпервване на времена или периода, а следващите OC – 4 са изх. имогат да регулират честотата

port B е еднопосочен и се състои само от изходи

намира се на адрес 1004

изходи ($A_8 \div A_{15}$) при разширен режим

Port C - той може да се използва и като вх и като изх
-което се определя от DDRC
Това е в едночипов режим

Port D

SPI и SCI - интерфейси
Двата старши бита са свързани с адресния строб $\overline{AS}$
Ако имам 1 - четене, а 0 запис - R/W

Port E

аналогови входове за ADC

Използва се за аналогови входове на ADC (АЦП), може да се четат и като цифрови вх.
Защитен транзистор
Аналогов мултиплексор с ключове
Ако го използваме като аналог входове, транз. T_2 е запушен.
Ако го използваме като цифр. вход, тоди транз. се отпушва

Аналогов мултиплексор

<u>ADC модул</u> – 8 битов АЦП – грешката ± 1/2 от най-малкия бит. Точни сф. Гарантира се линейност. Време за преобразуване е 32 такта, за 2MHz определят s. време преобразува 0 до 5V. Може да работи от 2,5 ÷ 5V

Входна характеристика

Паразит. капац е под 2pF
Защитен елемент, който е поставен за защита на входа
Аналогови ключ, който включва 20pF – вх. капацитет

Вр за включване 6 такта (3μs). Проблема е, че за 6 такта този кондензатор да се зареди до вход. стойн. т.е нашата времеконстанта да е по-малка от 3μs, Rизх < от няколко kΩ

<u>Блокова схема на аналогов-цифров преобр.</u> (12)

4-режима – един канал да измери 4 стойности; да мери 4 канала непрекъснато

На входа има аналог. мултиплексор M. УАП е цифров. От SAR има интерфейс и резултата може да бъде запомнен в 4 регистъра ADR1 ... ADR4

Когато работи 1 канал АЦП работи и ги записва в 4 регистри

Управл. на АУП става с 2 регистъра – OPTION – двата най-старши бита са свързани с АУП. С най-старшия бит се вкл. захранването, а с другия се избира дали да работи с тактова честота Е или да работи със собствения RC генератор

ADCTL

Младшите 4 бита са свързани с избор на канал.

M – едноканална работа, ако е 0 ще работи с 1 канал, а 1 – с 4 канала

C/S – 0 – едно преобразуване, 1 – непрекъснато преобраз.

CF – флаг, показва, че преобразуването е завършило

Това АУП може да преобразува само 1 канал. Да направи само 1 измерване и да спре или да направи серия от измерв. и да ги запише в регистъра. Не се използва много, точност не по-голяма от 0,5%

(14) SCI – интерфейс асинхронен

– при него имаме винаги стартов бит. След това се предава D0 до D7 и D7 и се предават 2 стопови бита. Предаването става с такт на такт. честота

Работи се на 16 шахта, за да не се вкара грешка и работи с тактова честота, хаото да делим на 16.

Да има възможност за предаване на D_8. Деветия бит D_8 е да се направи връзка м/у отделните микроконтролери и за да се различи адреса от данните. D_8 за предаване на адрес

Блокова схема на сериен интерфейс

TDR - 8 битов регистър, процесора пише данните и се прехвърлят в TDSR. Той е двойно буфериран и докато се предават данните можем да пишем други данни. От този регистър можем само да пишем

RDR - от този регистър можем само да четем

SR - статус регистър - регистрира състоянието - по него следим

CR1 и CR2 - контролен регистър

RG - делител за получаване на тактовата честота

CR1

6 и 7 бит са свързани
Ако искаме да предаваме 8 бит в 4 де записваме 1

Работата с 9 бит става с CR1

CR2

TE и RE разрешения за работа на предавателя и приемника

RIE - разрешение за работа на транслитера

SR

Отделните битове имат отделно предназначение
7 бит - показва, че предавателя е празен
6 бит - предаването е завършило
RD

OR - препълване, свързан е с приемащия регистър
NF - има шум
FE - флаг, който показва, че няма стоп бит

Две примерни програми за работа
<u>Инициализация на SCI</u>

```
LDAA   #$30        - задава се 9600 бита
STAA   $102B (Bavd)
LDAA   #0
STAA   $102C (CR1)
LDAA   #$0C
STAA   $102D (CR2)
```

данни→A

```
OUT   LDAB SCSR  (SR)
      BITB #$80
      BNE  OUT
      STAA SCDR(TDR)
      RTS
```

Данните са записани в акумулатор A

За да работи тази програма трябва в TxD, PD1 да е изход, в противен случай програмата няма да работи.
След Reset се прави ∀ портове да са входове

SCI

Чрез (9^та бит) можем да адресираме тези контролери
Винаги имаме стартов бит, който е 0. Това са данните, а ако предаваме адресен бит това ще изглежда по следния начин, 9 бит 0 това е адрес, а при 1 са данни

SPI - синхронен периферен интерфейс - предават се данни от D0 до D7 и синхронно с тях се предава тактова честота, по преден, по заден фронт

Винаги едното у-во е: Master- управляващо у-во а Slave- управлявано
MCU- прави запис в регистъра на Master-а и данните се прехвърлят в регистъра на Slave и неговите данни се прехвърлят в регистъра на Master.
Четене няма. За да прочетем пак трябва да запишем в Slave, прави се фиктивен запис. Master-а е активен, а Slave пасивен
Този интерфейс е удобен, защото има много АУП, SPI ⇒ ADC, DAC, упр LED

Ако трябва да управляваме 4 сегментни индикатори ни трябват 8x4 - 32 цифр. изхода
Изходите може каскадно да се свързват последователно
Много често индикаторите се намира там, където не е контролера
Този SPI интерфейс позволява да си поставим индикатори на различни места

Блокова схема

При инициализацията изходите трябва да ги дефинираме, като изходи

Инициализация SPI

```
LDAA   # $2F
STAA   PORD
LDAA   #$344
STAA   DDRD
LDAA   # $50
STAA   SPCR
```

38 = 00111000 (3, 8)

D_5 D_4 D_2

(13) Таймерна система на HC11

Блокова схема (с изх)

→ системи за генериране на времена

регистър свързан с разрешение за прекъсване

Output Compare – блокова схема (за 4 модули от TOC1 до TOC5)

Нед. програмно непрекъснато трябва да пишем в Register

(дали сме разрешили работа с прекъсване)

За работа на Output Compare системите и използват следните флагове

Всеки модул има собствен адрес на прекъсване

напр. TOC1 → FFE8, 9

По отнош. на измерване на f и T

Input Capture - системата

регистъра е свързан с брояча TCNT

ICxF - флаг

Ако искам да измерим период по предин фронт получава се строб Л дава възможност да се запише содер. на брояча в регистъра X_1

$X_2 =$

$\Delta X = X_2 - X_1$

Грешка ± 1 дискрет

За по-висока точност използваме брояч с по-висока честота

Напр. HC11 → 2MHz ; OSC 8MHz → 0,5μs

PIC → 5MHz ; OSC 20MHz → 0,2μs

MSP430 → ; OSC 8MHz → 0,120μs

Пример на програма за един елемент, който мери период

```
стои и чака → BRCLR  TFLG1, X $04
              LDD    TIC1, X
              STD    FRSTE
              LDAA   #$04
              STAA   TFLG1, X
стои и чака
нарастващ фронт → BRCLR  TFLG1, X $04
              LDD    TIC1, X
              SUBD   FRSTE
              STD    PERC
```

Тя работи с първи модул

X - показва индек. адресация

TFLG1 - проверява 2 бит от флага, ако е 0 тя пак чета, като стане 1 зарежда акум. A с рег. и чете содържанието от него като го записва в TIC1, X

- от D и изважда първата стойност и и записва в нашия период D не следващия ред

Real - Time

Тази система трябва да е тактувана, честота на дискретизация.

Трябва да следим процеса. Такива с-ми, които следят зад. процес периодично. Тактуването на с-мата зависи не само от микропроц. с-ме, но и от обекта на управление. Това е един брояч, който с разрешени през опред. време може

да дава прекъсвания

6 бит - флага се вдига при достигане на времето

- със 6 бит разрешаваме работата на тази с-ма

002^{13} 1 ms

112^{15} 32 ms

Real-Time FFF0 1 | програмно | брояч

Pulse Accumulator - брояч - 8 битов

Този брояч може да бъде директно свързан на входа PA7. Има и втори режим да броим тактовата честота

Схема в разширен режим с HC11 - порт C се превръща в мултиплициран порт за адрес

Система в микропроцесорен режим с HC11

Работата на процесора е такава, че при E 1 има обмен с външната с-ма

На MODA и MOD B трябва да има 1, за да имаме разширен режим. На port C е мултиплексирана магистралата.

Двете памети са с организация по 8 бита. Адресите от A_0 до A_{12} трябват да бъдат подадени към двете памети паралелно. $A_{13} \div A_{15}$ да ги подадем на дешифратор. За да можем да раздерем адресното поле. Остава само WE - който е за запис. WE ще получава като R/W през инвертор и подава на и ще получава импулс, който по заден фронт

Блокова схема на EVB (от упражнението)

EVB го използваме само да изследваме АУП, таймера

EVB може да го използваме за изследване на нашата с-ма в едночипов режим

Адресното поле да има адресна и потребителска страница

Ако ни трябва външно сериен модулатор използваме EVM

(15)

MSP-430 - програмен модел, основни модули

16-битов MCU с богата периферия т.е вкл. освен стандартните RAM, Flash памет и др. и включва АЦП, сериен комуникационен интерфейс, I^2 интерфейс

Имат два 16-битови таймери, аналогов компаратор чрез който може да се направи АЦП.

Работи от 1,8V до 3,6V, консумацията е 250μA /на MIPS

Има 6 режима на намалена консумация т.е 6 low power

Честотният диапазон е 32кHz ÷ 8MHz

125 ns - време за една инструкция

Предназначен е за портативна апаратура

Програмен модел на MSP-430 - включва 15 регистъра, които са 16-битови и са означени с R0, R1 ... R15 и имат възможност да се използват и за адресиране

R0 - PC - програмен брояч

R1 - SP

R2 - SR - статус регистър

R4 ÷ R15 - с общо предназначение

x 31x ÷ x 33x – са СБС

EPROM памет, малка консумация и работят от 4,5V ÷ 5V

F11x1 – с Flash памет

F14x . . . . . –"–

АЦП при тези две е 12 битово

F4xx – има Flash и драйвери за управление на течнокристални индикатори, които имат малка консумация

Блокова схема:

JTAG – да се програмира Flash и да се дебъгва. Прогр. става по канала Usart (първия)

Osc – вграден в CPU – има три генератора; това осигурява висока сигурност

Progr. памет – Flash – движи се от 4k bait ÷ 64k

RAM памет от 256 ÷ 2k байта

Вх/изх. портове – 5 на брой 8 битови порта (P1 ÷ P5)

Usart – серийни интерфейси – могат да работят синхронно и асинхронно

Comp – аналогов компаратор – има делител и опорно напреж, изх. му е прекаран през инвертор чрез което може да се премахне Offseta.

TB - таймерна с-ма - (MODTA, MODT B) - модул таймер A и таймер B

WD - таймер - осигурява работа на следене на програмата

Схема на Osc

I - осцилатора може да бъде високочестотен или ниско честотен (~32 kHz кварц)

II - осцилатора може да бъде само високочестотен 1÷8MHz

Схема на аналогов компаратор (Comparator A)

На изх. на компаратора има инвертор и ключ който се превкл. едновр. с ключовете пред компаратора

Целта на компар. е да се унищожи Offset. като това се постига с две измерв. и от едното се вади другото. Позволява лесно да се трабет с RC-групи измервания

MSP 430 F1121 – в 20 изводен корпус

Съдържа: Flash 4кВ

+256 date

256 байта RAM

Port P1

Port P2

Comp. A

OSC.

WD

Таймер А

(17) Въвеждане на аналогова информация

фиг 1

Начини на свързване

Как да свържем АУП към микропроцес. с-ма – 1) директно ~~на~~ чрез магистралите – предимства: най-високо бързодействие, най-бързо се обслужва АУП. Изисквания за такава организация

Недост – за да работи така АУП, то трябва да има изх с високоимпедансно състояние (тристой)

2. Чрез използване на SPI интерфейс – може много лесно да подменяме едно АУП с друго

SPI позволява АУП да е изнесено на разстояние от микропроц. с-ма. Позволява с-мата да е разсредоточена

Нед – данните се предават серийно и обмена е много по-бавен

3. Да сложим паралелен интерфейс (чрез порт) – Пред – нямаме проблеми с високоимпедансното състояние

Персонален компютър (PC)

Слот модул пред: - може да използваме захранването на PC
Нед: - вътре в PC е много по-шумно и ако искаме да измерваме сигнали с малка амплитуда (μV) няма да стане.
Реално чрез слагане на модул, може да отпадне гаранцията на PC

Чрез RS232 има модул и подава
Пред: чрез RS 232 за въвеждане на инф - стандартен вход
Нед: не може да се предава с висока скорост на серийния интерфейс

Принтерски интерфейс - данните се предават паралелно на 8 бита
Чрез него можем да вкарваме генер от
Скоростта е много висока

USB - освен двата сигнала идва и захранването

Предимства на използване на PC в сравнение с вградена микропр. с-ма.
1. Трябва да има място за прогр, RAM, I/O, трябва да има часовник за реално време, възмож. при изключване на захранв, трябва да и запомни инф, възможност да се визуализира инф, Неправено специално за конкретния случай ⇒ много висока цена.

2 PC - да въведем аналог инф, запомнящо у-во има, печатащо у-во, памет има, можем да визуализир инф. - монитор, меню програмно осигуряване

Такива с-ми много лесно се проектират на базата на PC

От какво се състои един такъв модул

Сензор - температурен, за влажност, сигналът от него е много малък

Pt_{100} - 100Ω при 0°C платина 100

0,385 Ω/°C

При платината се пуска 1mA и 385 μV/°C

При термодвойка - константан 37 μV/°C - чувствителност

Този сигнал може да се усили 100÷500 и да се филтрира

V_{ref} - на базата на него преобразуват АЦП - V_{ref} - няколко волта

Тези с-ми са многоканални - пред усилвателя има мултиплексор чрез който се превключват отделните сензори.

Изисквания към тези системи - 1) галванично разделяне

Управляващата МПС е на разстояние от сензорите и може да се получи потенц. разлика м/у масите. Желателно е при индустриални условия да се раздели масата на сензора от масата на МПС т.е галванично разделяне - чрез галванично изолирани ОУ.

Разделяне на цифр с-ми с оптрони - b° стабилност е много голяма в сравнение с ОУ. Нед: - бр. на оптроните зависи от начина на предаване на инф. Може оптроните да станат 10÷20бр и те имат консумация.

2) Шумоустойчивост на вх. сигнал – честотната лента на ОУ.
Зависи от избора на АЦП – изменение на t^0 е бавно – АЦП може да е с двойно интегриране, който е много шумоустойчив и не е нужен филтър.

3) Избор на АЦП
ADC с паралелен интерфейс
ADC с SPI интерфейс
– с I²C
По отношение на унф. интерфейс – с поразрядно кодиране – за бърза АЦП.

16 бита ADC 1100

Проводяване

Съвременните АЦП са проектират с отделни пинове за аналог и циф. маса и отделни захранвания. Неправилното проводяване води до големи смущения

Свързващия проводник има някакво съпротивление. Тези пикове ще вкарат смущения
Аналог и цифров. част се проводяват поотделно и това спр. на цифр. проводник е отделено и из бъ свъ

смущенията от цифр. част да се вкарат в аналоговата част. Във захранването се слага някакъв филтър и през захранването да не се получава влияние върху аналоговата част

ADS 1100 – АЦП със серен интерфейс

16 битов

На входа има усилвател, който може да усилва 1, 2, 4, 8

2,7 ÷ 5,5 V

I_{cc} = 90 μA

8 измервания

Това АЦП позволява много малка консумация, малък корпус

сигнала е от 1 ÷ 2 mV/V

При захранване 5V ⇒ 5mV → 1kг

5μV → 1гр

С-лите от тензомоста са много слаби

Единия начин е подаваме V на моста и получаваме разбалан на моста. За да се намали грешката от t° дрейф захр. се с 5 V усилва и и се измерва и се обръща захр. и се измерва и и отчита дрейфа като разлика

AD 7730

АМ- аналогов мултиплексор, превкл. вх. сигнали

DAC- ЦАП

ΣΔ ADC- има много сложен филтър

АС - тактов генер - за обръщане на захранването за емитиране на дрейфа.

АЦП на микроконтролерите

68HC11 → ADC - 8 битов 16μs

ADC MSP 430 F149

ADC

MSP 430 F149

На входа имаме аналог мултиплексор. АМ, чрез делител може да се мери захр., и чрез диода t°.

Вр. за преобразуване е по-малко от 10μs

16 регистъра по 12 бита в него се записват резултати

16 управл. регистри

АУП, като е вградено и е 10 битово.
Бр. за преобразуване 20μs
2 пъти по-бавно

ADC при DSP (унф. процесор)

Има разлика от 430:
АУП е 10 битов, но е много бърз 500ns
унф. блок за резултати.
За управл. се мерят 2 тока еднобр.
Имаме управл. регистри за двата резултата и като се пусне АУП да измерва токовете и да ги записва в регистрите.
За да се разтоварят DSP от управл. на АУП.
Разлика - възможност по 2 канала да управляваме АУП.

20.05.2003г.

фиг 1. +

MSP 430F149
MSP 430F 169 + DAC (12)

За да използваме МП система аналогов с-л трябва да го превърнем в цифров
За да го измерим трябва да го прекараме през DAC.

Наличието на УАП позволява да изведем на екрана на осцилоскопа с-лите и да ги наблюдаваме. Така се следи даден процес (управлява се даден процес). Това са системи в реално време

Тема № 19. Извеждане на цифрова информ. и нейното въвеждане

Цифрова информация

1. Съгласуване по ниво – ако имаме две ИС работещи съответно на:
   - 5V
   - 3,3V

изхода има екв. схема

за да ограничим тока слагаме Rогр.

За да има съгласуване

а) се слага последов. резистор с който се ограничава тока.

Но това понякога създава проблем.

Можем да поставим RC група, която създава паразит. капацитет, който да доведе до закъснението на фронта на импулса.

(намалява бързодействието)

б) с делител $R_1 - R_2$

SMT 160 – температурен сензор

Когато имаме обратният случай се слага само един резистор да защитим нивата и да ограничим токовете

Когато е отворен колектор - слагаме един резистор към +5V
Понякога се практикува ниу 12 и 24V - съгласуване със
схеми със последователно свързани резистори
то лесно се съгласува изхода откъдето входа

2. Шумоустойчивост по време

1.
2.

За подаване на един. импулси се използват ключове, с пружини за да направят контакт
Тъй като контакта с пружинката се удрят два контакта се получават малки вибрации
Това може да се избегне като се сложи RC-група, която да ги интегрира тези пикове Тази група да е по-голяма, но трябва да се съобразяваме с τ
Прави се едно закъснение от 20ms. и след това се чете.

Генератор на единични импулси може да стане с RS тригер

3. Индикация - когато имаме някакъв микроконтролер към него има много цифр. вх и измизат цифр изходи
От гледна точка на подържане е важно кой вх и изх. е задействан, поради което на вх и изх се поставят светодиоди

светодиоди, за да се види, кой вход е задействан

4. Полярност на входните сигнали

Така трябва да си проектираме схемата, че да нямя значение полярността на входния сигнал

5. Галванично разделяне – слага се оптрон

Цифр. входове да се реализират с галванично разделяне – чрез оптрон. Тази схема грец ни осигурява да не се интересуваме от полярността. Зенер 8,4V по отношение на амплитудата, U-s под 8,4V няма да протече ток. Чрез оптрона осигуряваме галваничното разделяне

По отношение на цифр изходи

1) Галванично разделяне

Въвеждане на информация

Ако няма натиснат бутон всички входове са 1. Има схема "И" на която входовете са 1 ⇒ и изх е 1. Щом на един от входовете има 0 (на сх. И), то на изхода на схемата "И" ще излезе нула и ще дава заявка за прекъсване. Прави се бягаща нула, за да се установи кой от бутоните е натиснат

Предимство – използва се само един аналогов вход за въвеждане на цифровата информация

Въпрос №20 Визуализация на информацията

Тя е много важна. Възможностите са:

- светодиоди – по-ярка, но не достатъчно
- седесегментна индикация – тя е* светодиодна и с* LCD (течнокристална) → колкото е по-силна светлината толкова по ясно се виждат, проблема е, че при $T<0°C$ не се виждат, даже и на тъмно

* видеодисплей

а) векторни

б) растерни

Един от начините за управление
През Port да се управлява статично индикатора

Седемсегментните индикатори се използват със специализирИС
които имат Clock вход и данни

Motorola MC14489 - може да управлява 5 индикатора
Тази схема позволява да се управляват 25 светодиода
по отделно
LCD дисплеите са течно кристални индикатори с матрица
Принципа е че се подават със съответния чип вътре.
LCD- За да се изобразява 1 символ имаме 5 на 8 точки

Това не е най-доброто. Използва се една матрица 5x8
Чрез тази матрица се изобразява
1 символ с опред. точност

Блокова схема:

Контролера има 8 битова магистрала DS - вх. на
който може да се записва
на два входа (по единия
или др)
V0 - трябва да е аналогово
се регулира контраста напрежение, най-често с потенциометър
Консумацията е 1÷2 mA. - много малко

Имаме два начина на управление
1) активно
2) с изчакване на 20ms, т.е през 20ms
се записва по 1 bit.
R, W - сигнали са ни необходими, защото не трябва да правим
много записи, за да не се губи информация

Индикация:

ИС-ми, които реализират динамичен тип индикация, ИС имат вътре регистри, сериен вход, сериен изход и 8 изхода, с които могат да и управляват светодиодите на сегментите. С една ИС можем да управляваме 4 числа

Допълнително - контролера се намира на едно място, а оператора на друго и използването на тази ИС можем да я поставим където си искаме. Може много лесно да и да и свържем допълнително

Друга схема МС 14489 - принципа е същия. При първата е нужно да се сложат транзисторни ключове, а при втората тези ключове са вградени, може да и управлява 5 индикатора

Може да и управлява ~~се~~ 4 сегмент отделно, а при първия не е възможно.

Векторен дисплей - при който лъча се движи в двете посоки по x или y. При тях не се използва електростатично отклонение, а Електромагнитно отклонение - отклонението е ~ на тока

1. отклонителна бобина

Необходимо е по x и по y да се подават непрекъснато напрежение

Разширен дисплей - ако това е екрана лъча се движи като описва хоризонтални линии като след всяко се гаси

хоризонтални синхрони импулси

активно време на просветване

вертикални синхрони импулси

⚠ Увеличаването на разрешаващата способност намалява времето.

100Hz → 10 ms 10μs

Тази развивка, която се използва в телевизията се е приета за мониторите

Как става изобразяването на буквено-цифровите символи?

Имаме знаков генератор, който представлява постоянна памет - там където е 1 ел. лъч просветва, там където е 0 ел. лъч е загасен

	1	2	3	4	5
1	0	0	1	0	0
2	0	1	1	0	0
3	0	0	1	0	0
4	0	0	1	0	0
5	0	0	1	0	0
6	0	0	1	0	0
7	0	0	1	0	0
	0	0	0	0	0

празно поле, което да раздели редовете

5x7 най-грубия знаков генератор

Обикновено се използва 7x10 или 8x8

С колкото повече точки се работи то изображението е по-точно

Представлява памет на която се подава адрес ASC II

Как се реализира изобразяването на инф?

1. Изобразяването на 1 буквено-цифров ред става едновременно

~~1 2 3~~

Обикновено един монитор

монитор
1000x1000 - точки
100Hz
10μs
10ms
> 50MHz

Цветна TV

Използва се на базата на R, B, G

в зависимост от комбинацията на яркостта на светене на всяка точка, получаваме отделните цветове

В графичен режим ни трябва повече памет от колкото при буквено-цифров. Днес основно се работи в графичен режим

DAC - контролира яркостта за получаване на нови цветове
DAC се намира във видео картата

Комплексен видео сигнал

(21) Интерфейси

При захранване с 5V нивата са му TTL

I токов кръг - интерфейс дава възможност да се предава информация на големи разстояния с ниско бързодействие (в момента не се използва). Използва се до 1000 до 1500 m

I около 10mA - 20mA
R - са защитни, за да не се подадат 5V директно към транз.

наличието на ток е лог 1 - транзистора е отушен
не наличието на ток е лог 0 - транзистора е запушен.

II RS 232 - приложим е в комуникациите

RS232 е напрежителен интерф.
лог „0" се предава над +5V до 15V
лог „1" се предава -5 до -15 до -25V

RS 232 - има 9 пинов и 25 пинов

В max 232 има преобр. на ниво с които се получават нивата за RS232

III RS 485

RS 422 - диференциален интерфейс, включват 10 приемника
Той е едно посочен

12м 1200м
10м 100к

RS 485 - диферен интерфейс с нива ± 1,5V
Позволява двупосочно предаване

На базата му може да се изгради локална мрежа (32 предавателя, 32 приемника могат да бъдат свързани)

Този интерфейс се използва в промишлеността
12 метра → скорост 10MB

SN75176

Предавател, който преобр. ТТЛ нивата
PT - разрешение за предаване
Използва се в промишлеността, защото е двупроводен

• Супресори - ограничава U в двете посоки

5,8V → 1mA
6,8V → 10mA
10V → 38A (≤1ms)
15V → 174A (≤20μs)

Зенера се използва за стабил. на U
Елемент, който може да поеме много-голямо натоварване по мощност, но за кратко време се нарича супресор

В зависимост от нивата на линията съществуват различни супресори

27.05.2003

CAN интерфейс (сериен) - предназначен за автомобилостроене
Подобен е на RS485 по отношение на нива и буфери
Удобен е и се използва в индустрията
SPI - синхронен интерфейс (сериен)
I^2C интерфейс (по сложен), но позволява да се реализира в мрежа. Може да се реализира програмно, но скоростта е по-ниска

Разлика м/у SPI и I²C:

1) Има start за четене и запис
   SCL - такт. честота
   SDA - данни
   Първо подат данните в нула
   след това SCL в 0 - при запис
2) stop импулс - първо тактовия сигнал в 1 после сигнал данни в 1
   R/W = 0 - запис
   R/W - 1 - четене

При SPI се изпраща цялата инф. наведнъж

Може да се свържат до 8 устройства към този интерфейс и чрез трите байта да адресираме главното устройство работи с всяко поотделно
След stop данните са в 1, или във високо импедансно състояние, които се подържа от резисторите

<u>Паралелен интерфейс</u> - използване за въвеждане и извеждане за инф в

* за принтери (Центроникс) в персоналните компютри
- той е 8 битов, (по-висока скорост)
- по-висока скорост

В началото е бил само за извеждане, но сега е двупосочен

Използва се за връзка на далечно у-во с PC

Тема №22 Интерфейси на PC

ISA - 8 битов ISA - първия интерфейс в PC (по-прост)

PCI - - дъчна платка (industry standart architecture)

заявки за прекъсване

DMA0 ÷ 3 - 4 канала

↓ бърз запис

I/O адресно поле за PC

0000 ÷ 000F - DMA контролер

0020 ÷ 0021 - програмирай контролер за прекъсвания

0040 ÷ 0043 - 8253 - програмируем таймер

0378 ÷ 037F - паралелен интерфейс

03F8 ÷ 03FF - сериен асинхронен интерфейс

Допълнителна логика за дешифриране на адреси A0 – A19

трябва тази модул да е във високо импедансно състояние за да не пречи

За поставяне на АЦП или друго у-во са нужни допълнителни блокове за дешифрация осигуряващи сигнали за четене, запис и т.н.
Адресите не се подбират произволно, а само тези, които са свободни и не пробват на самото РС

(23) Настройка и диагностика на MPU системи

Диагностика със статични въздействия

Диагностиката

* диагностика по време на разработка (статични тестове)

– проверка на хардуера, дали той отговаря на точното проектиране
Проверка дали паметта отговаря на съответния адрес
Няколко етапа на настройка
Тест дали на отделните адреси се обаждат съответните устройства, както трябва да са там, доколко хардуера е проектиран и изпълнен правилно
– проверка на връзките
– проверка дали има прекъснат проводник или писта
– програмата (софтуера) на системата се пуска модулно (на части) за да бъде изчистването на грешките максимално бързо и ефективно
Това, че статичните тестове минават не значи че системата ще работи – например може да има закъсняния на сигнали или объркване на сигнали.

MPU може да се замести със с-ма от превключватели, но тя ще е много обемна и сложна, по този начин се проверява работата на с-мата.
Може веднага да се провери дали адресацията е вярна.
Така се проверяват връзките и може да открием дали има късо или прекъснатата (късо) писта.
Програмата трябва да се пуска по етапно за да се провери по лесно

Логиката след дешифратора не винаги може да се провери, тъй като проверяваме статично. Ако има проб и състезание м/у сигналите не може да се провери работата

за статични тестове

Схема за проверка на дадена логика

Поставят се ограничаващи резисторите имитират буферите, които управляват магистралата (изходното им съпротивление)

(24) Тестове с динамични въздействия

Дава възможност за тестване на харда и софтуера

Вътрешно схемния емулатор - обикновено има монитурна програма, която ни дава възможност за тестване. Те са основното развойно средство за разработване на микропроцесорни системи

Предимства:

- можем да спираме програмата на различни точки

Недост - вътр. сх. емулатор трябва да има ресурс

ROM емулатори – една RAM памет в която чрез интерфейс може да се зареди програмата и след Reset тази програма да се стартира

Нед: При такава памет и спадат трудно точки на прекъсване

Пред – ROM емулатора се прилага независимо от микропроцесора

74HC... – буфер за MSP430

DSP Цифрови процесори → фамилия C2000 – тези процесори първоначално са били моцно АЛУ и регистри за извършване по-бързо на операциите

За целта в рамките на една инструкц. има две обръщания към паметта

PIC при 20 MHz за 200ns се изпълнява една инструкция

(1) 40 MHz при 25ns – TMS320LF2407A (3,3V) (16 битов)

(2) TMS320LF2810 (1,8V) – работи на 150 MHz → 6,7 ns на една инструкц. (32 битов)

CPU на (1) – е 32 битово, което позволява по-бързото извършване на изчисления, разполага с допълнителни регистри и таймерни системи, имат системи за генериране на честоти, имат изходи за управление на мост ключ

Включват ADC (анал.-цифр. преобр.) – 10 битово и преобр. за време 500 ns

Има Flash памет памет до 32к думи

RAM памет до 2к има и DRAM – двупортов

RAM

Периферия – SCI – асинх. интерфейс

SPI – синх. интерфейс

CAN – интерфейс

възможност за работа с JTAG

DSP:

- Имат високо бързодействие, но за сметка на голяма консумация (100мА), захр 3,3V
Тези DSP служат за индустриално приложение, когато ни е нужно голямо бързодействие, висока изчислителна мощност
Те не са много по-скъпи в сравнение с другите микроконтролери

QEP - тази логика е вградена
Напр: инкрементален датчик вграден към микроконтролера, като една умна схема

(18) Извеждане

Галванично разделяне на УАП:

1) чрез усилване на изхода
2) чрез цифр. разделяне

УАП се свързва най-често чрез SPI интерфейс

I Чрез УАП: max 520 → 8 бита → УАП

II Чрез преобраз на f → U

промяна на ШИМ и чрез RC филтър

MPU 8071 : 8k Flash
640 date flash, 256 RAM
работи от 3 ÷ 5V, като при 3V консумацията е 3mA

TS - температурен сензор

(16) Блокова схема на TMS320LF240xA.

Ext шрит - логическа решетка

Таймерна система:

Бл. схема на АЦП:

Обща блокова схема