

14.02.2011

MCU

16:20

Упр от 16:30

2 чукала по 3 уп.

машинен код

1 протокол общу.

18.02.2011

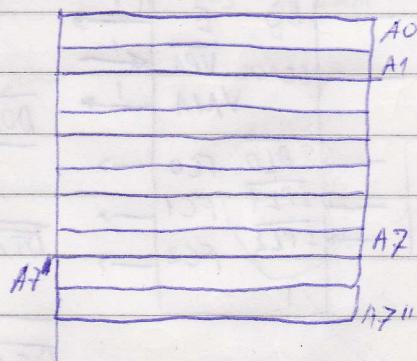
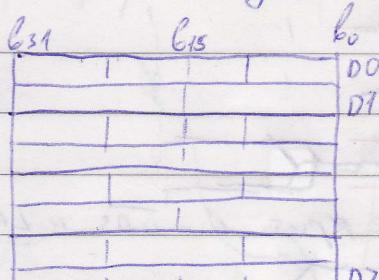
Лекция - MCU

6800

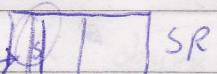
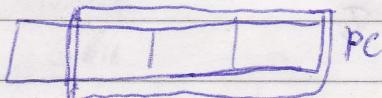
8x 32 bits регистора (общи) 16bit - АЛУ

24 bit - адресна магистрала

16 bit - даннича магистрала



данни рег.



A7' и A7'' - използват се като указатели на стека

S bit - работи в системен режим или потребителски.

- За да може упруга система да използва ресурсите (дактиите) на процесора е необходима система за арбитриране

- Reset може да бъде и бх и иск.

18.02.2011г.

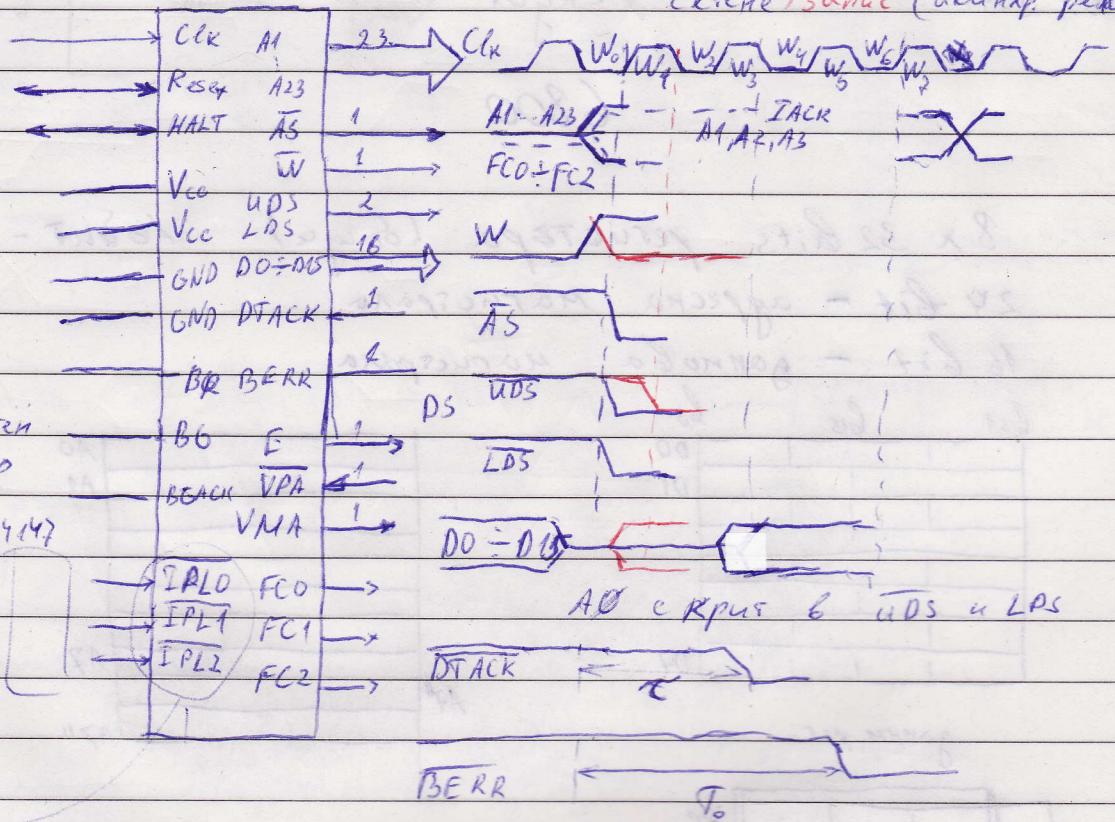
Модуларна за обмен на данни.

68000 е асинхронен процесор

DS - data strobe

A1 -- A23 - адресни линии

читане/запис (асинхр. режим)



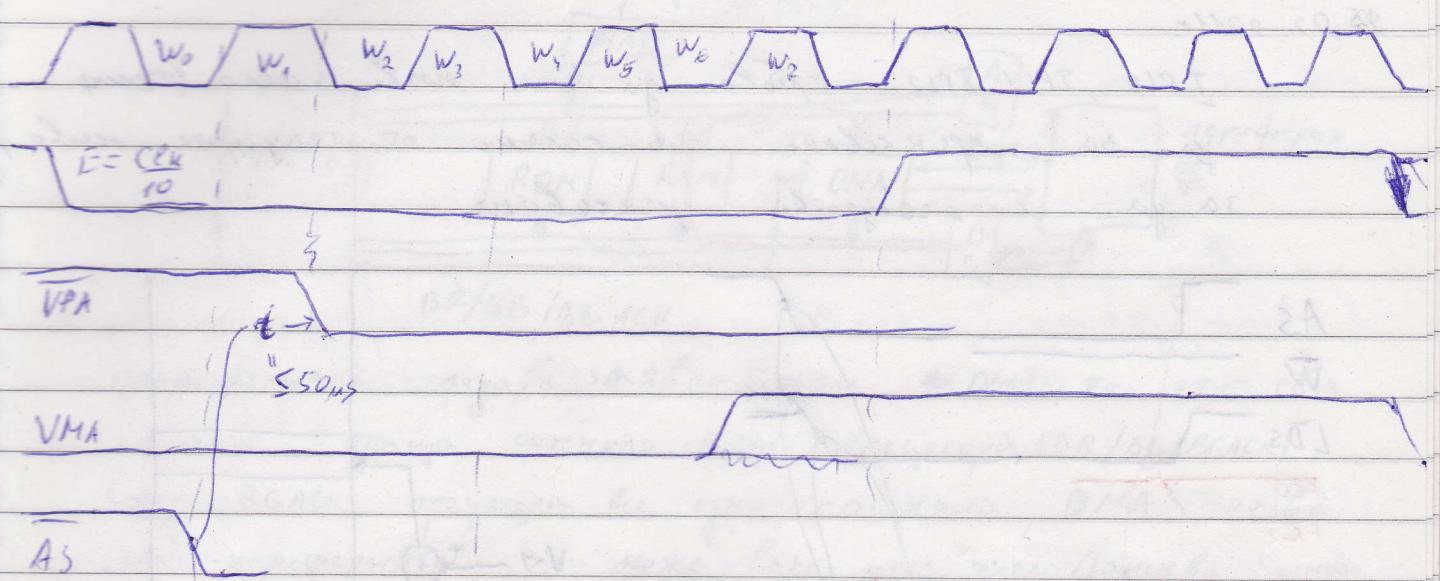
прекъсване

BERR - сигнал за прекъсване на текущата логика времето
в случаи, че не се осъщува DTACK

-R-M-W - Read-modify-write

#AS - сигнал с "0" а R и W са взаимно синхронизирани

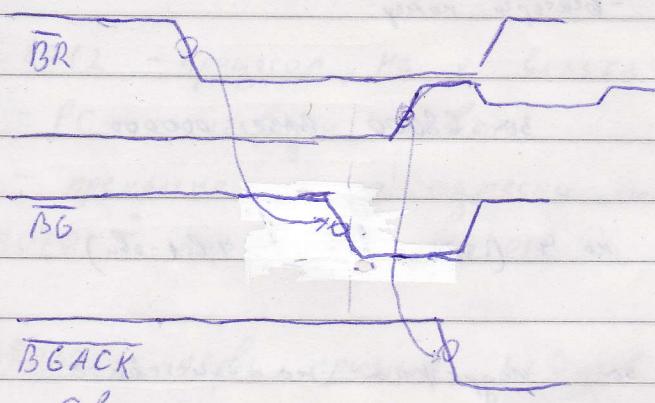
сикропен ресурс



BS - bus request - запрашивается
ресурс

Адресирване за гочен

BG - bus grant

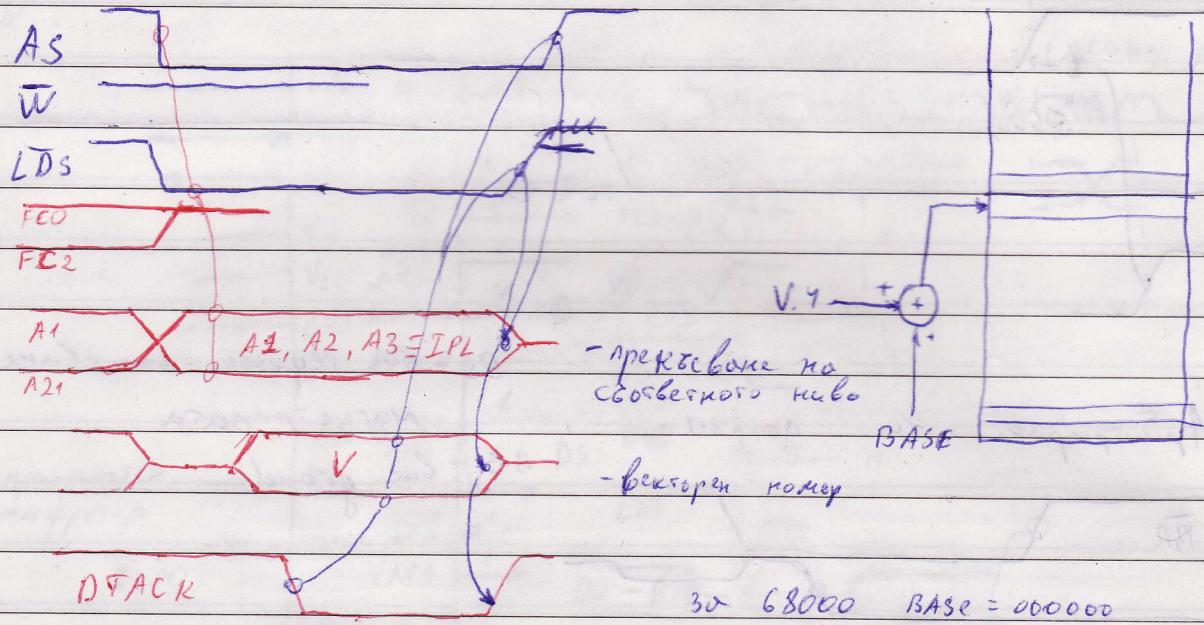


$\#S$ bit

FC2	FC1	FC0	FC - обрън. ког
0	0	0	- 139pbc
0	0	1	- DATA, USER
0	1	0	- ADDR, USER
0	1	1	- предпб
1	0	0	- предпб
1	0	1	- DATA, SW
1	1	0	- PROB, SW
1	1	1	- JACK - interrupt

25.02.2011г.

IPL0, IPL1, IPL2 - граба да се нааби и усно флаги
за да пресъбаче по-горното от регистровото поле,
за да се осигури пресъбачане



Векторите се участват във 4 (так като са 4bit-ови)

Първите 64 вектора са за подгответа на системата
останалите 192 - user

Идея

- имитирате на линия A и имитирате на линия F - вектори
от първите 64.

- 68881 процесор

* тригонометрични функции

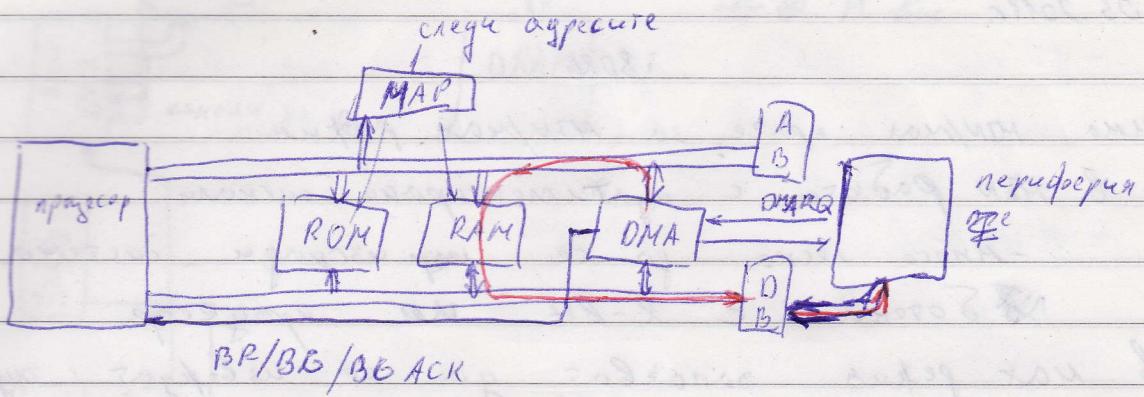
* 80 битова регистри

- Ако опкода започва с Axxx

* инструкцията ще се придвижи
с фиксирана запетая

- ако започва с Fxxx - малка запетая

запетая



- периферията подава згвжики като DMA, то от своя страна прифа сигнал REN на процесора (BR/BG/BGACK). След BGACK процесора все едно го казва, DMA разва до периферията, за да може вече да тя го записва инфо.
- 68451 - устройство за упр. на паметта

XC68012 - процесор, но е базиран в чиповете серийно производствени

- PC е изведен на бази
 - преколко много адресни линии згвжи не се е използвал.
- 68020 - клонинг на 68012

68020 = първи промишлен 32битов на ~~Motorola~~ Motorola

- Аритметичният процесор работи като ко-процесор
- асинхронен обмен по данни с допълнително бързодействие проверка за синхронизъм. За разлика от 68000.
- има синхронен обмен между данни

68030

- устройство за упр. на паметта е бързодействие 68020.

11.03.2011r.

18086

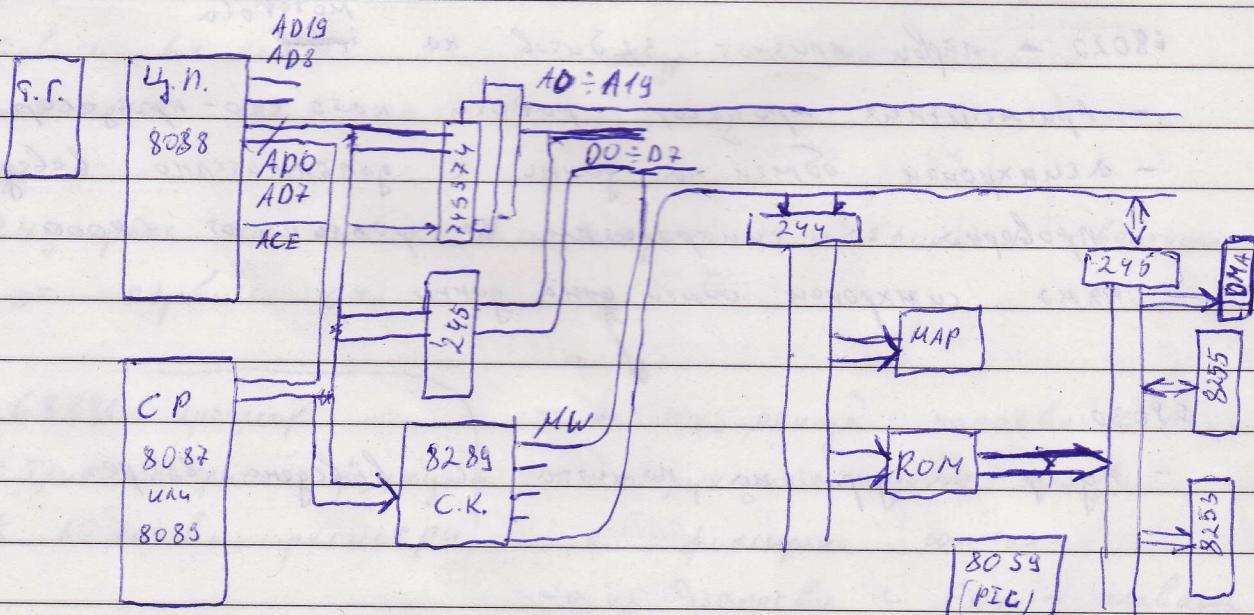
- има MIN/макс крате за MIN/макс резултат
 - в MIN работи с мултиплексорни сървюри
 - много лесно да се изгради система изпълнителна база с 8-bit или процесор.
 - в MAX резултат започва да се извеждат още един стартус код и показват с системния контролер
 - 8086 е първи 16-bit процесор.
 - при 8088 допълнителна магистрала е 8-bit, но различката между брзодействието на 8086 е само 14%.

808* и 8089 - KO-процесоры

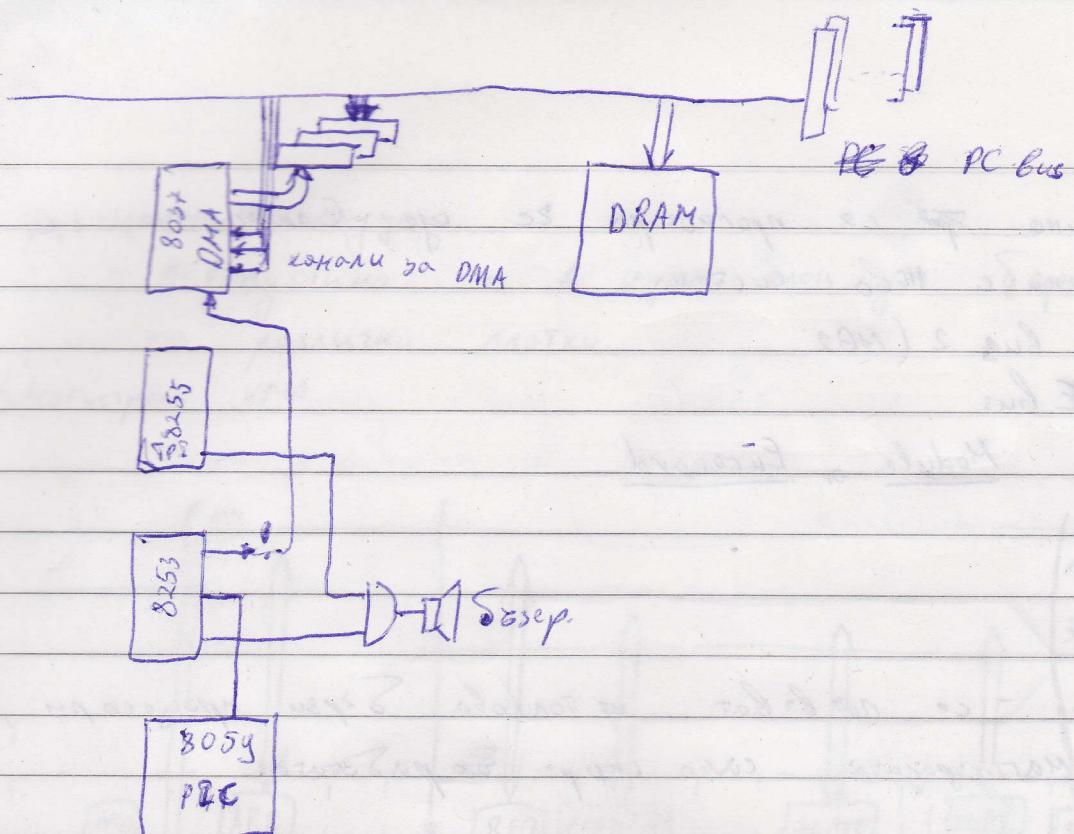
8089 - I/O co-processor

PC

CP - CD - processor
ЦП - центральный процессор
Т.Г. - тактовый генератор



8253 - Rainier



~~VESA~~ PC bus \rightarrow ISA \rightarrow EISA \rightarrow VESA

18.03.2013г.

M - multi
instruction

SISD - single instruction single data

MISD - система от конвертиращ тун. Специализирани компютери.

SIMD -

MIMD - паралелни системи от тази и следващата лекция.

- Магистрално податъчна система

- 65 разширени системи CANAC, с които се осъществяват в едни и същи експеримент модули от различни производители.

- разделни мачти за зетене и занос (24 места)

- 85 година се прогнозира за епоха САИЧ епоха се
издели с нова система

- Multibus 2 (MB2)

- VMEbus

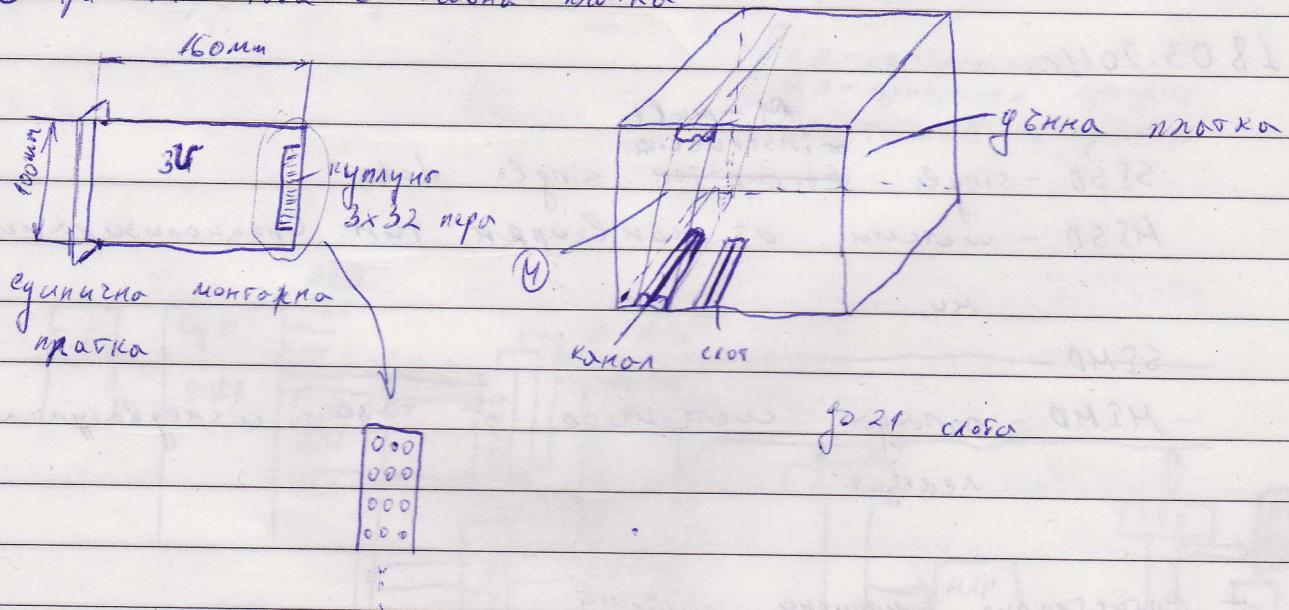
Versa Module on Eurocard

-> Verga

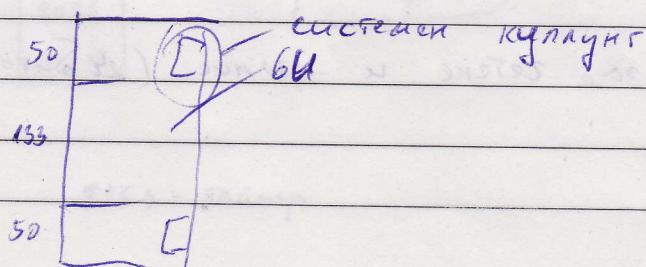
- 91-92 г. - се появяват прътъвата борзна процесори, за
VME материнската също сирия бордови.
- VME 64 - 64 битова

crate - ражера

№1) при VME това е наименование



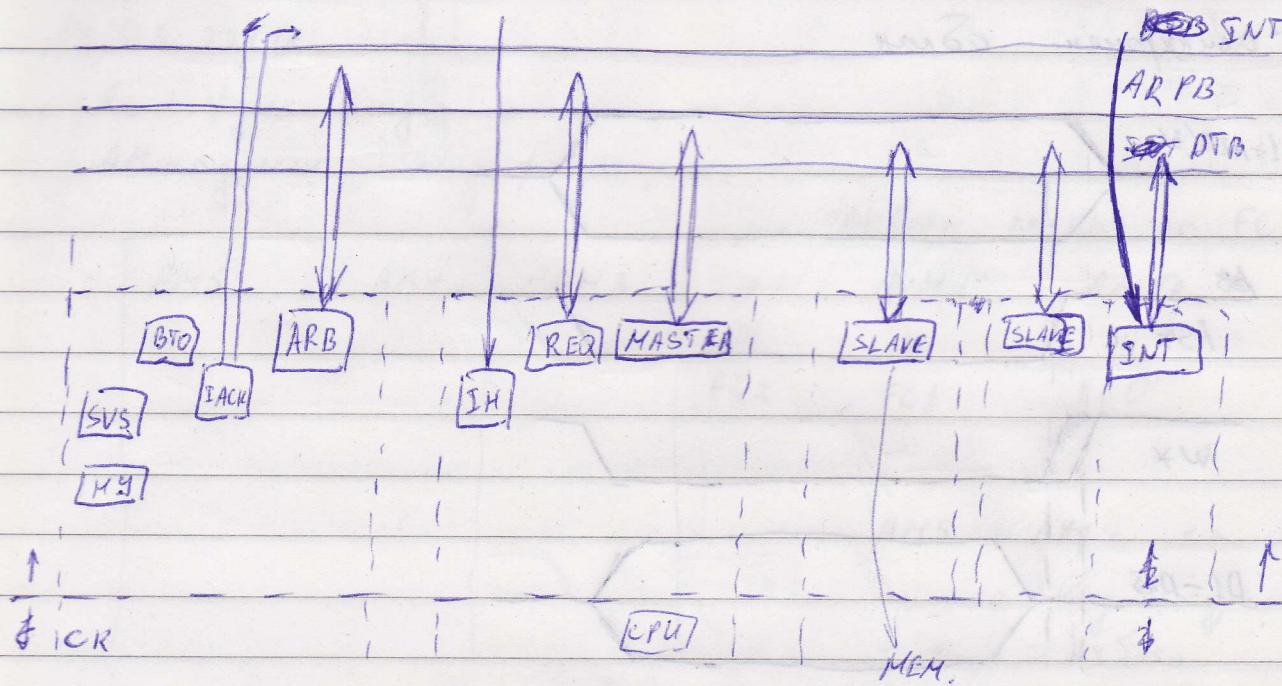
- За разшириване обемната е 133mm, плаката се
разделя на 2 и между тях се слагат 133mm



- функционалният модул

- Адресуемостта е в функ. блок да бъде разделян
на различни платки.

Логичният VME



кой пътвата позиция с с по-обеден статус. Там се
наима система контролер

MASTER - позиция се по това, че той дава адрес.

- може да има няколко SLAVE

- ARB - арбитър. Арибитрира достъпа

- IH - interrupt. Приема заявките за прериване

- SLAVE - с кого master извршва обмен.

IH - interrupt handler

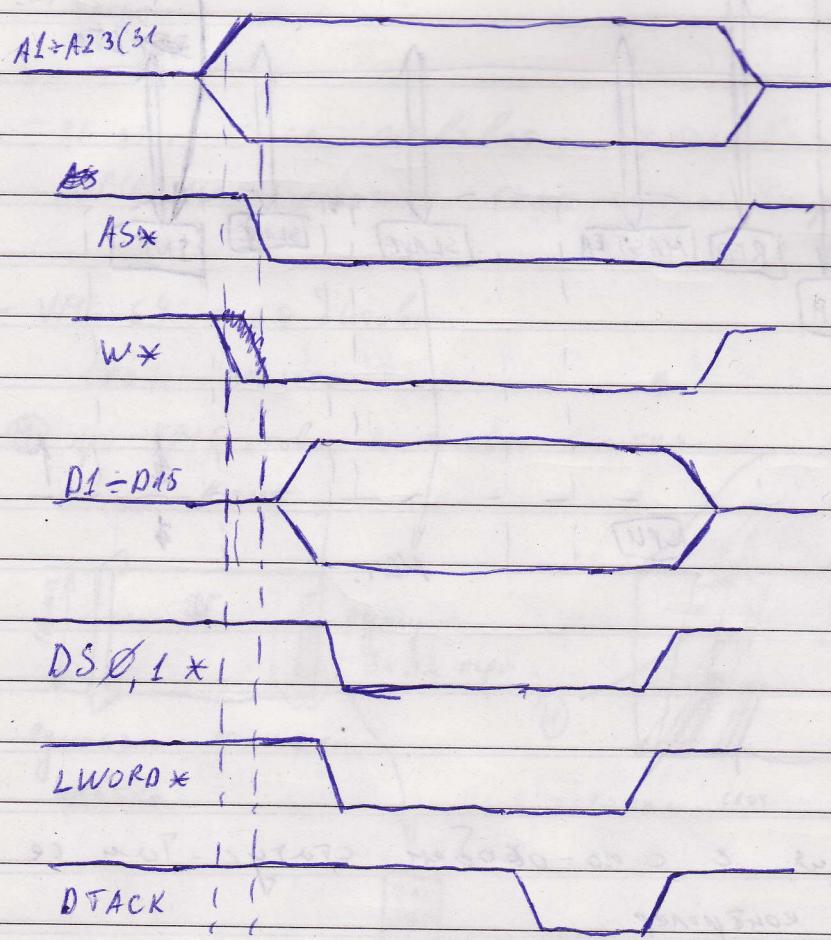
- когото има SLAVE като
процесор.

- 1 СК и ~~одно~~ общу за модула.

② Год система за обмен на данни

* - активното ниво е "0"

- асинхронен обмен



Осъществяване на блоков обмен

- при възникване на AS , адреса се запомнята
в SLAVE и от тук като запозна га се
работи с Data strobe като всеки модул аре-
ст в SLAVE се инкрементира с 1.

! Поставя се време за стабилизиране на адреса !

- при Versor Bus при всеки байт има по един байт за парити контрол. За разлика от VME къде е улеснено работата.

19.03.2011г.

FC - функция код

AM - адресен манипулятор

Извършва логика за FC

AM5	AM4	AM3	AM2	AM1	AM0
			111	111	111
			FC2	FC1	FC0

AM5 и AM4 за отработване
на за време адреса
- Stand = 24 часа

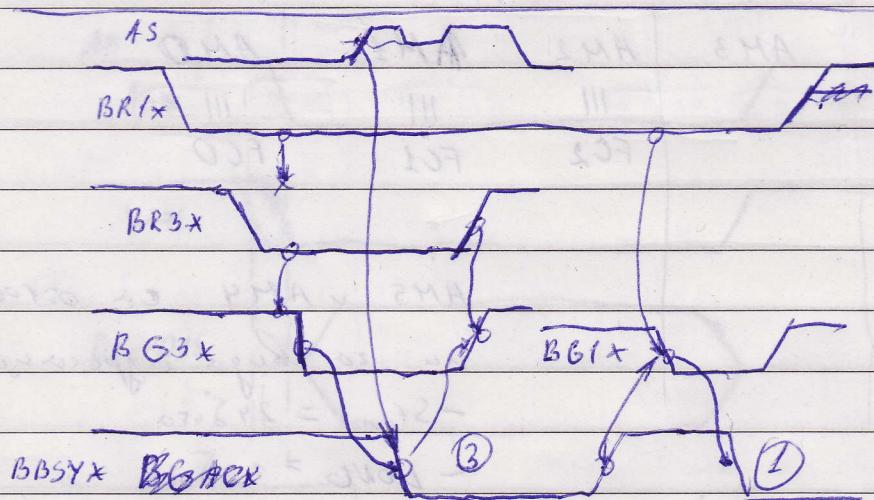
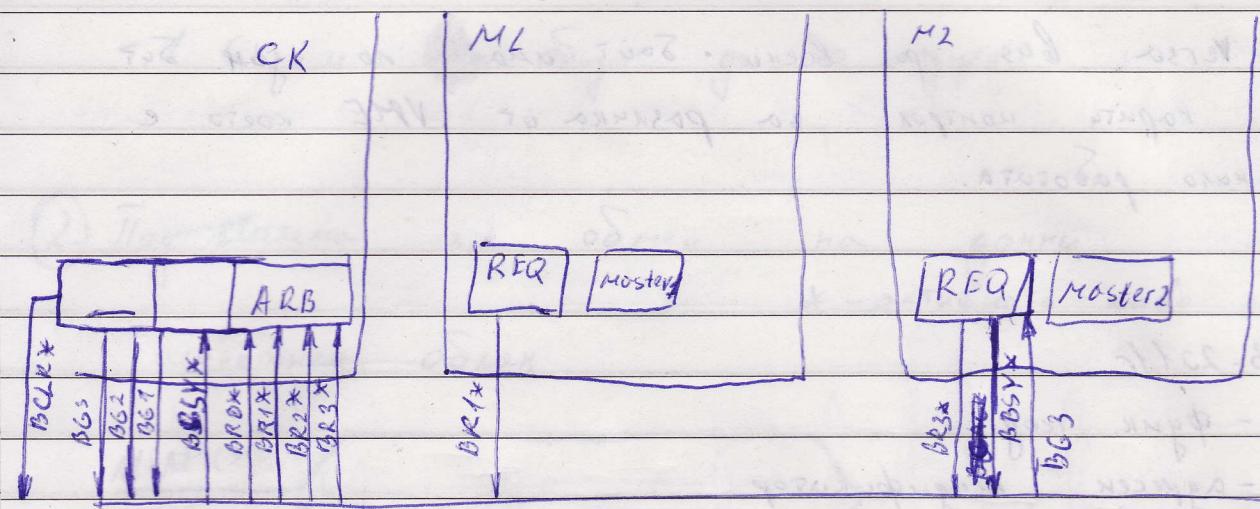
- LONG = 32 часа
- SHORT = 16 часа

AM3 докато е 6 в близко

когато дистанцията е близка, ако е
60 не е близка и това

както сме си съмнение от непредвиден

- Система за обработка



AKO yozige zashchitnoy ot BR2 a chez BR1x 6 momenta ARB negala curkan BCLRx

BB84

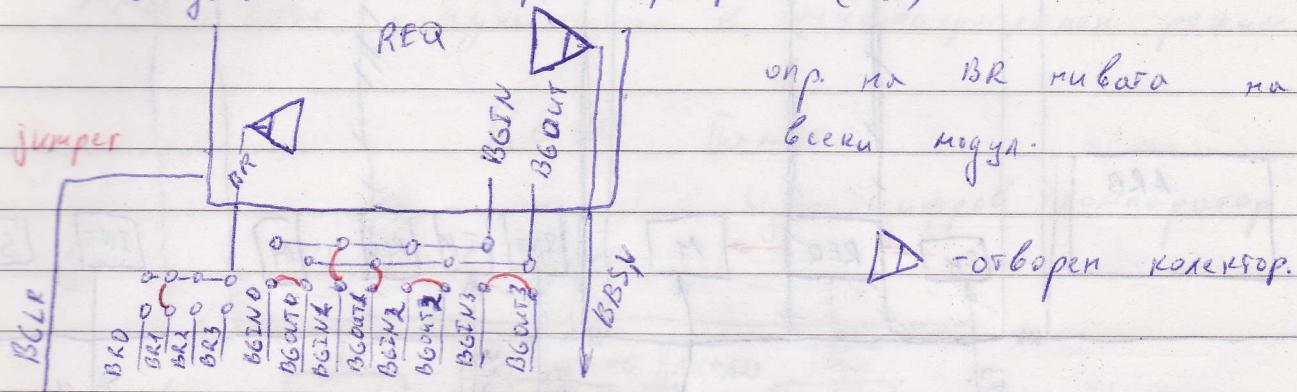
design chain - mogut na BGIN u BGOUT. BGOUT
ot CK chuzha 6 BGIN no i mogut ot tom BGOUT na
mogut go BGIN na bixx mogut.

AKO yozige prekobane BR1x u 6 u 2 mogut
na negolu BR1x - CK eye bixne BG na 1 mogut
a tol' cyc negoli BBSYx

- Onyce.

daisy chain е onyce - работа в приоритетна схема
~~(ONE)~~

- onyce с фиксиран приоритет (PRF)



- работи с 1 приоритетно ниво (ONE)

- ~~използва~~ се приоритетното ниво място

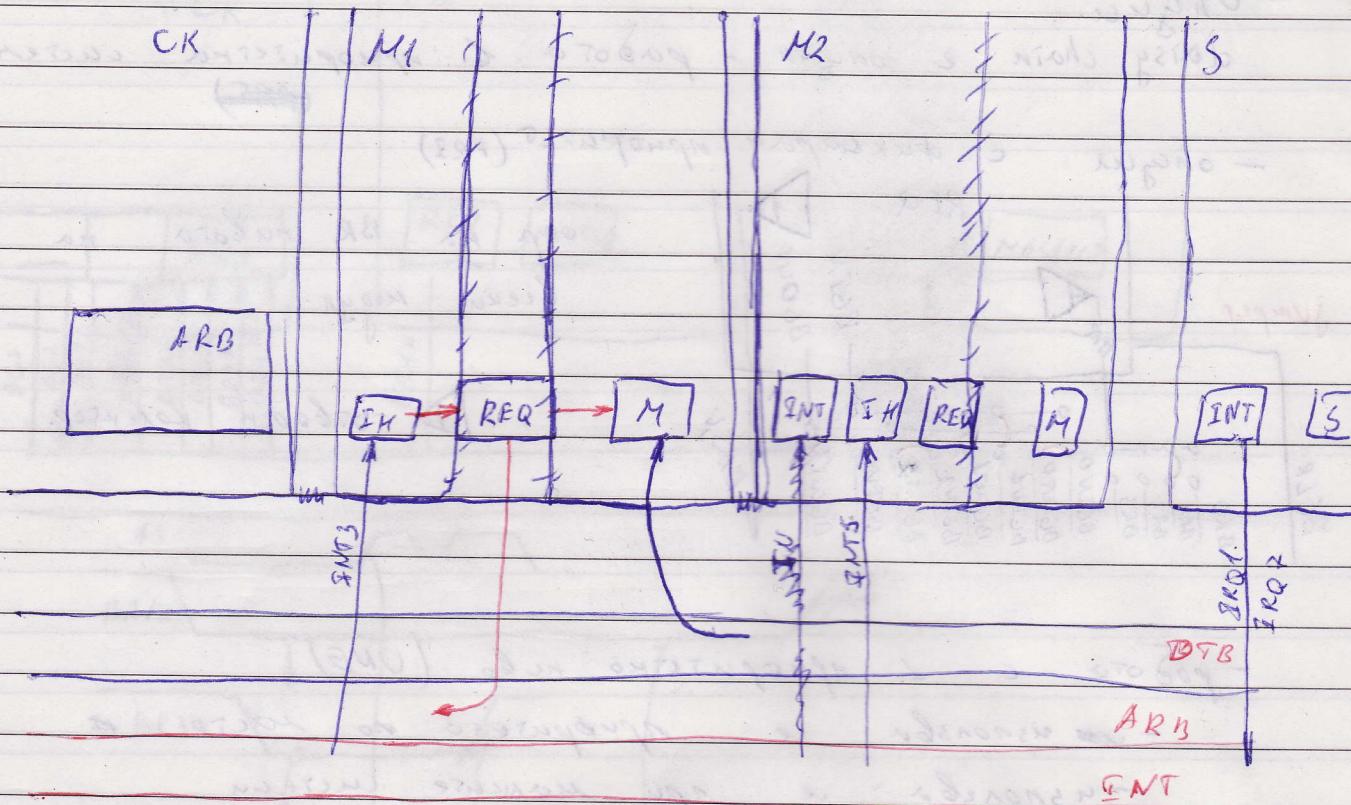
- използва се при малките системи

- RRB - уникатно съществува се приоритет.

- в нис-високия приоритет се е поместе 1
на всяка чарка, а останалите са забележават
с 1.

- Задължително

- на място работи замък за

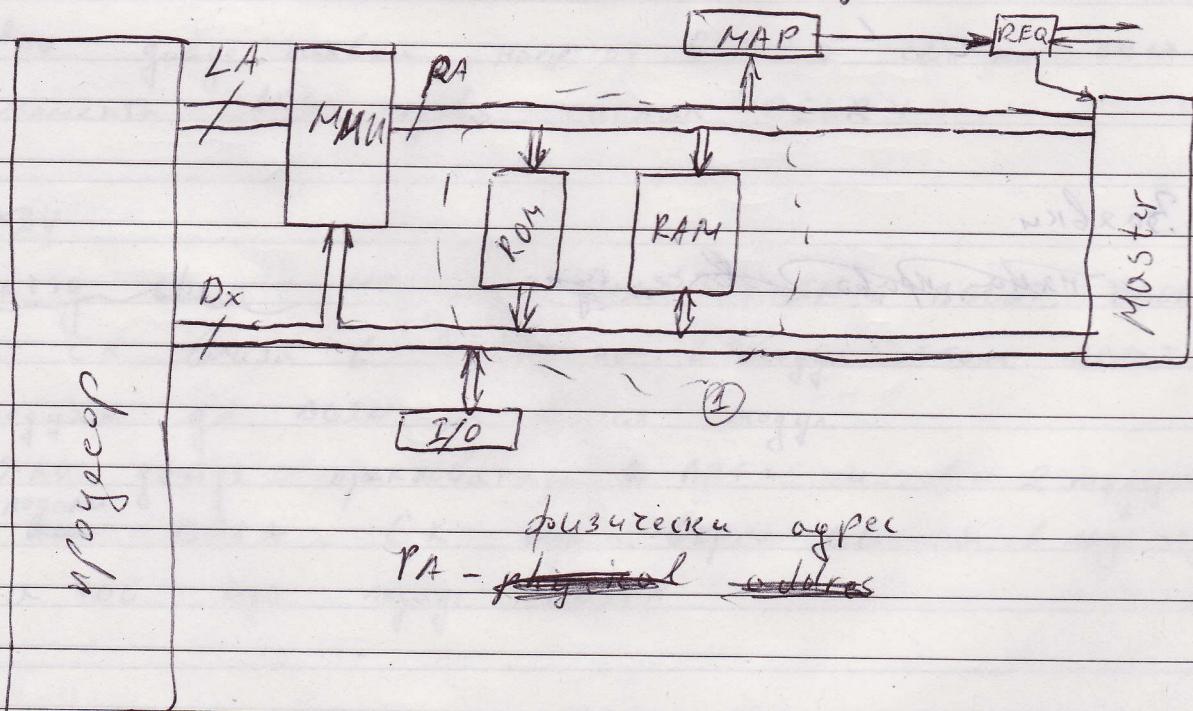


370

01.04.2011r.

MMU - memory manager unit

LH - логически агреси



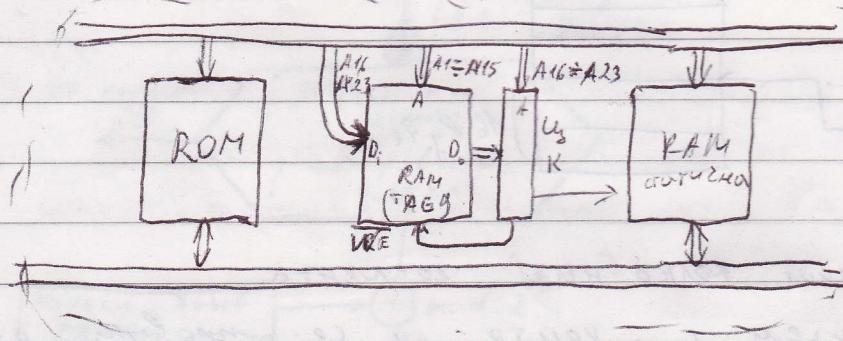
- MMU - ~~установка~~ бързо чита дескриптори
 - MMU - следи се за читане и за съхранение на данни

MMU и CPU комуникират в микропрограмен режим.

КЕМУ Помет

УК - унитровъв компаратор

① WE - write enable



~~32Kx8bit~~
16

32Kx8bit

CPU издава адрес пример.

03 15 F0

16:23 1:15

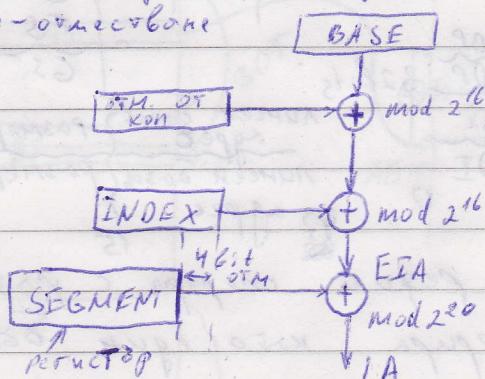
старши
20CF

младши
20CF

08.04.2011г.

регистри	6
AX	0
BX	
CX	
DX	
BP	
DP	
SP	
DI	
SI	
DS	
ES	

КОН-код на операцията



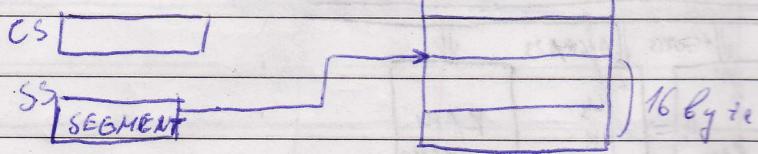
LA - линеен адрес

EIA - ефективен

адрес

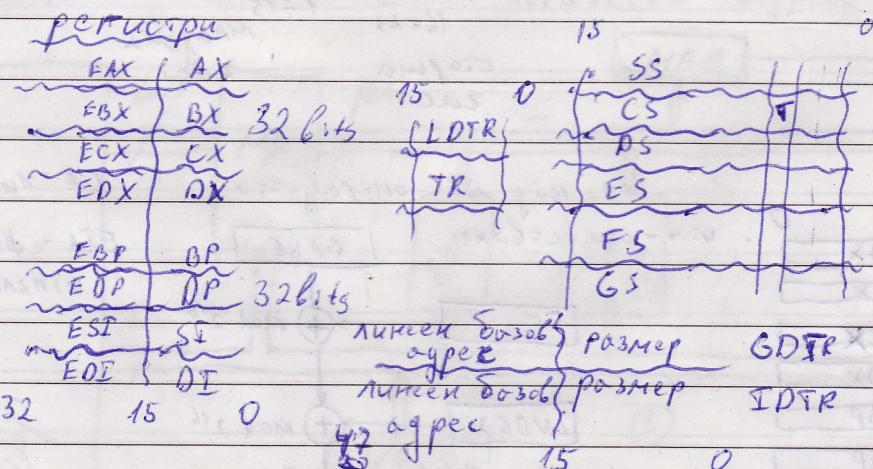
IP
FLAGS

SS - Stack segment



- имеа ограничение КОЛКО ИМЕЕ СЕГМЕНТИ.
- има механизъм с която да се провери дали има застъпване на сегменти.

процесор { 486 } !!!! |

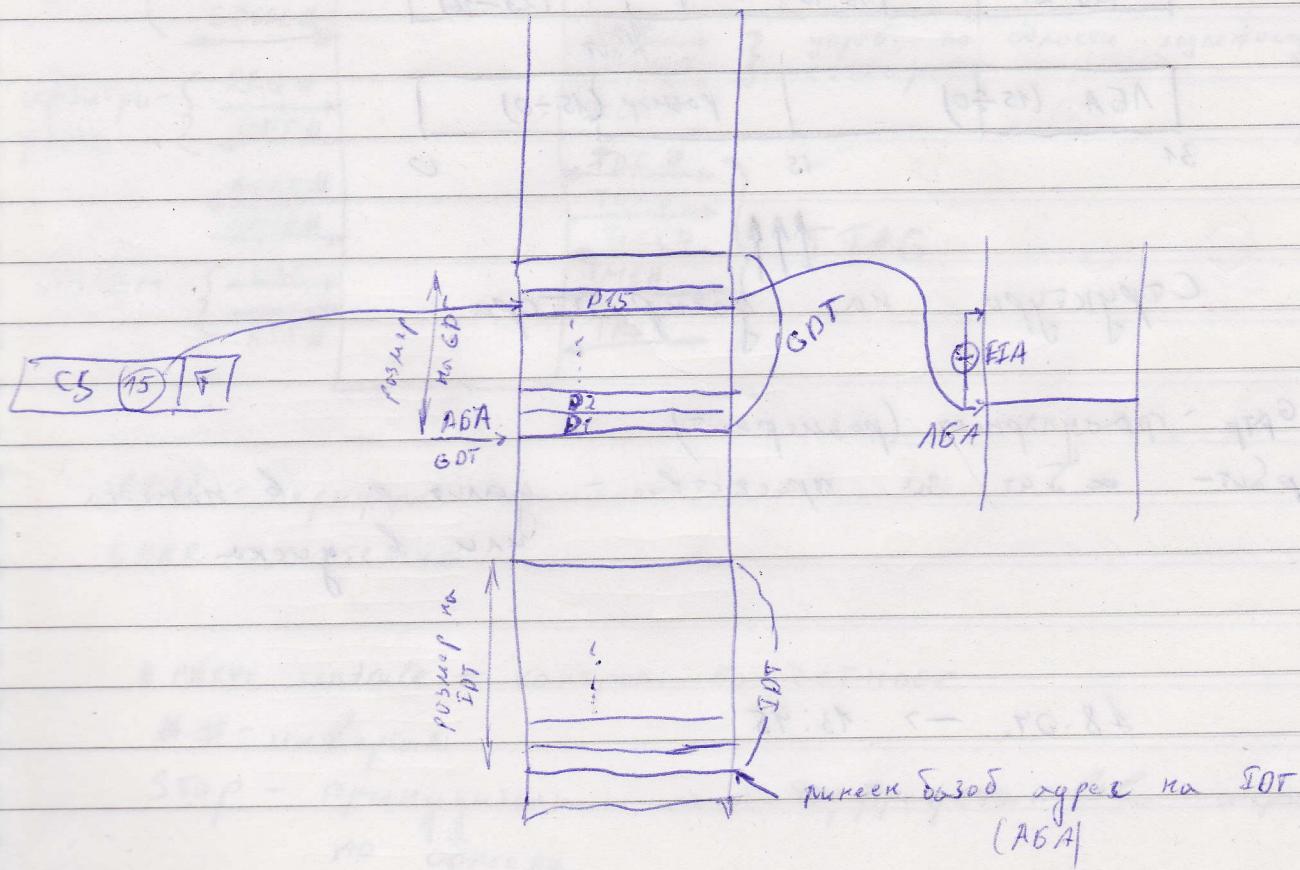
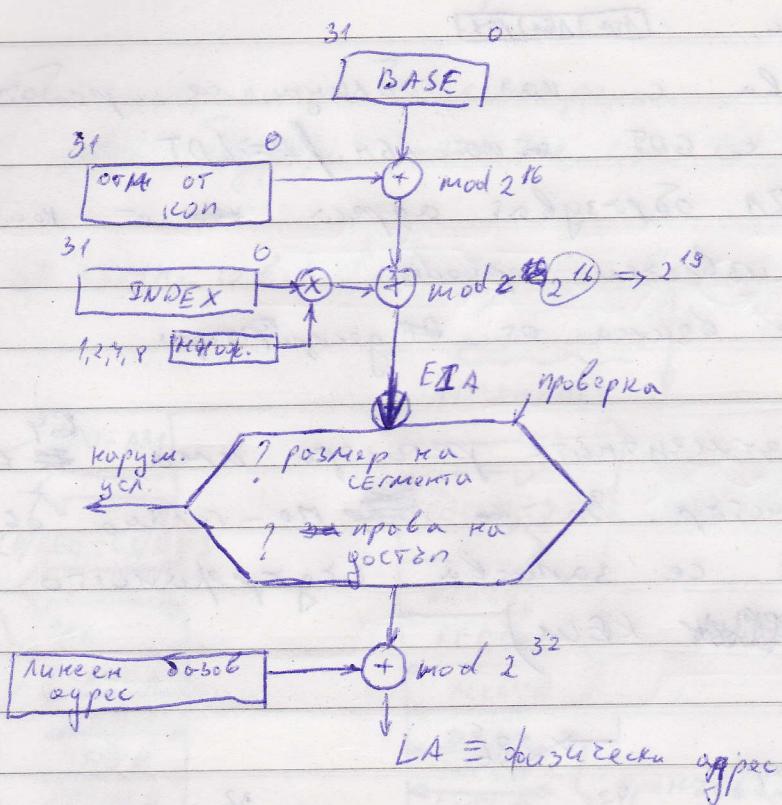


- б заличен регистър оперира с разширение регистри идикатора като едни без поддържан процесор

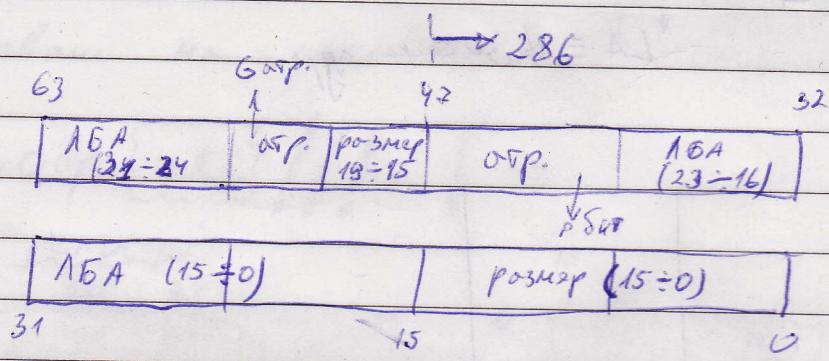
LDTR - регистър на логаритмична дескрипторна таблица
TR - task регистър

GDTR - global description table register

IDTR - ~~дескриптор~~ преселектор



- АБА и размера на GDT и IDT се генерираят в пъзелото.
 - Всеки дескриптор е записан в себе си АБА, размер и такива атрибути [API 1164] / 03
 - Дата "Г" показва с коя година се работи
 - АДО О с GDT ~~се~~ по АБА / ~~в~~ IDT
 - АБА + ЕИА обръзват адреса ~~от~~ от която трябва да извлечем адреса
 - АБА се винта от дескриптора
 - Всеки от регистрите регистрира ~~на~~ ^{въ} bit сегментните регистри имена ~~на~~ ^{въ} бит. За ~~се~~ ^{се} по-голямо разширение. В тях се записва ограниченията на дескриптора. (~~RECS~~ RECS)



Структура на дескриптора

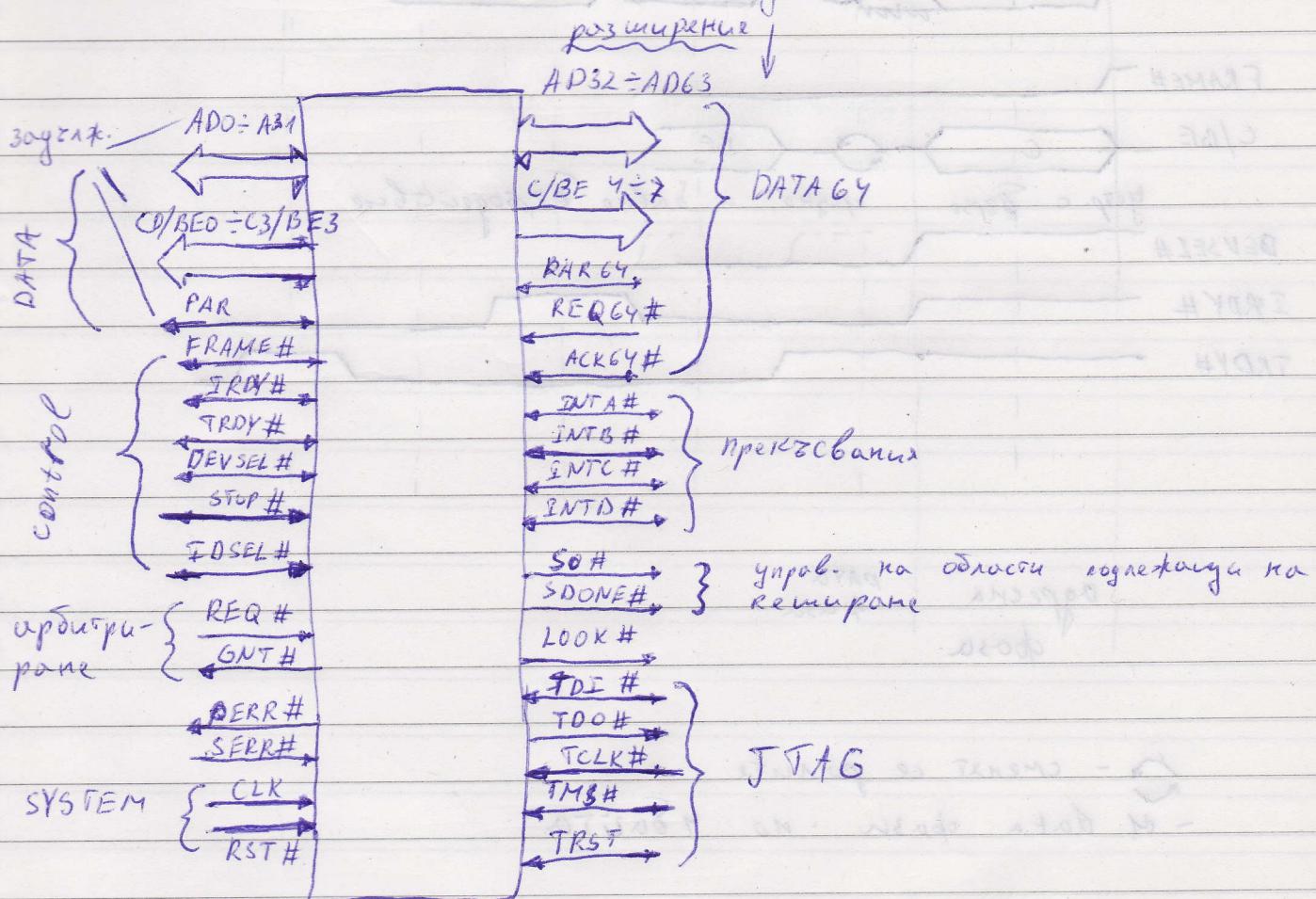
- ГРПР - группировка (размерность)
 - разб - разб за присстие - один в 6 номера или в 6 упак.

18-07. -> 13:45

15.04.2011г.

PCI - peripheral interface компоненти и интерфейс

- Магистралата е синхронна.
- Няма буфери (т.е. не се правят допълнително задавателни)
- ~~33 MHz, 32 bit~~ / по слоеве се еднат на 66 MHz



PERR - периферна грешка

SERR - системна

STOP control - контрол на зетност

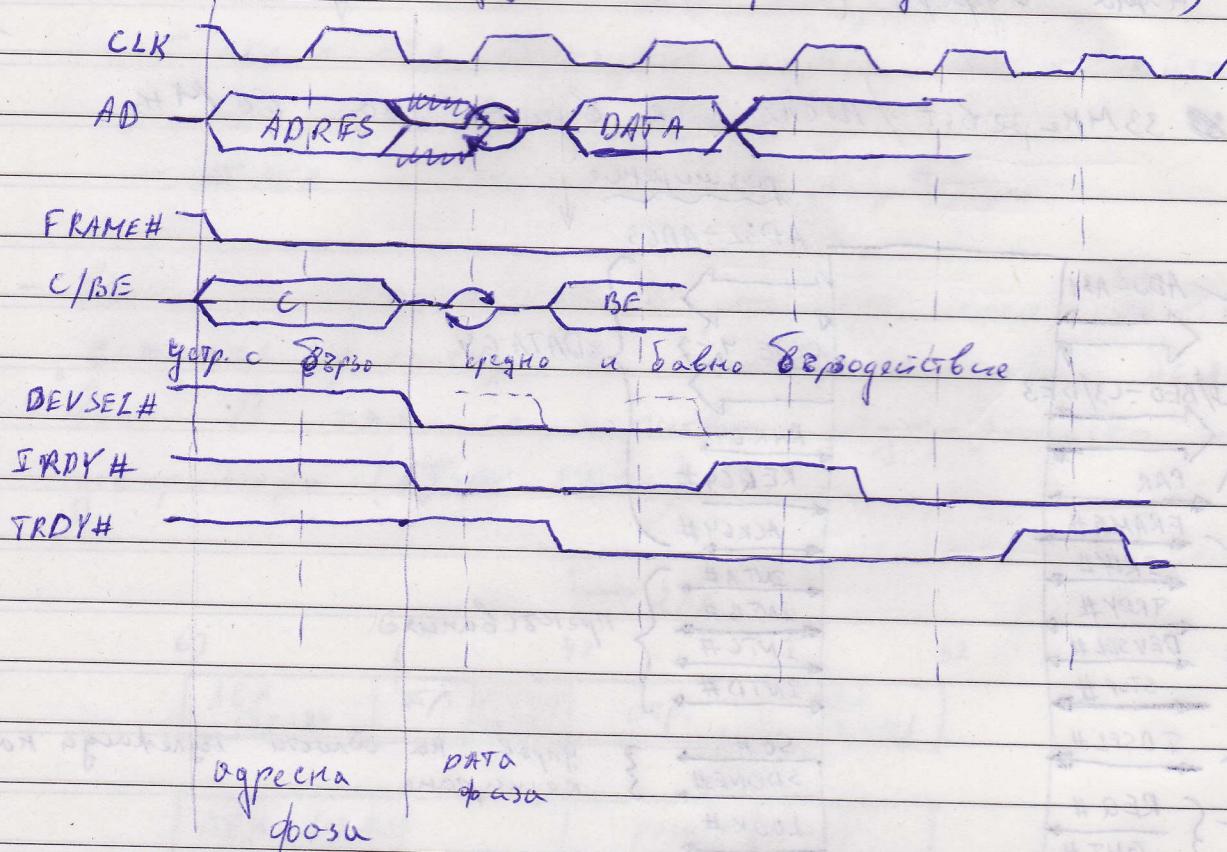
- инверсия

STOP - принудително се прекъсва работа операциите
no оримка

IDSEL# - инцинициализиран е за всеки един слот. Идеално
което десигнаторът.

DEVSEL - device select

C/B/E - както се изполви - R, W, interrupt
(номинална което грешка ще е изпълнена)



○ - сменят се по ред
- при data обрън по чинита

IDSEL# - 16 Bits - Направи ~~2 бита~~ 2 бура за
проверяване

STATUS - 16 bits