Lecture 24: 80x86 Bus Interfaces, DMA, and DMA-Controlled IO

Seyed-Hosein Attarzadeh-Niaki

Based on the slides by Barry Brey

Review

- 80x86 interrupts and the 8259 chip

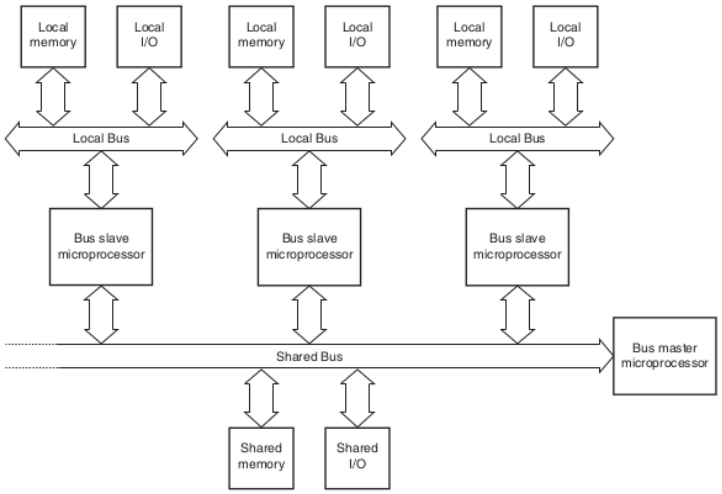
Outline

- Shared bus operation
- Bus interfaces
  - ISA
  - PCI
- DMA and 8237
- DMA-Controlled IO

SHARED BUS OPERATION

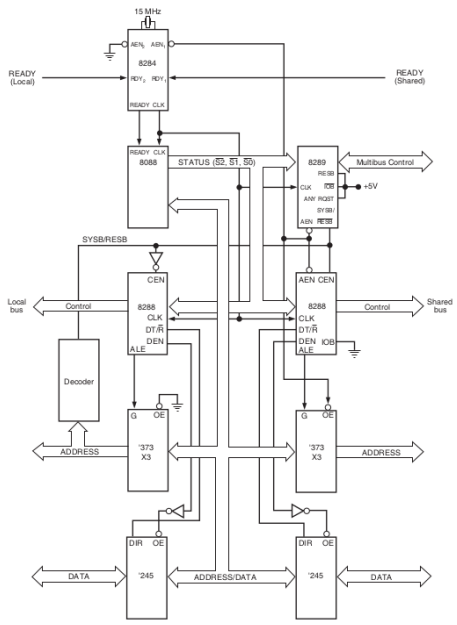
- In a distributed, multiprocessing, multitasking environment, each microprocessor accesses two buses.
  1. the local bus: connected directly to mem & IO
  2. the remote or shared bus: shared mem & IO
- 8086/8088 uses the 8289 bus arbiter for shared bus operation.
  - 80286 uses 82289 and 80386/80486 uses the 82389
  - The Pentium–Pentium 4 directly support a multiuser environment.
- The bus master is the main microprocessor in the PC.
- ISA bus is operated as a slave to the PC's microprocessor.
- The PCI bus can operate as a slave or a master.

SHARED BUS OPERATION



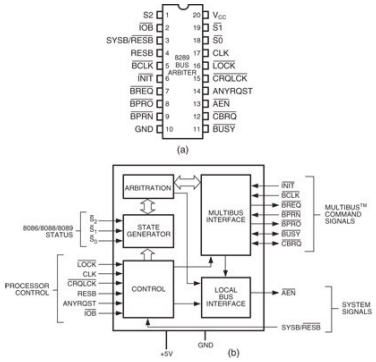
8088 connected as a remote bus master

- bus master applies to any device that can control a bus containing memory and I/O



The Bus Arbiter

- The 8289 bus arbiter controls interface of a bus master to a shared bus.
  - Controls the READY input of 8086
- Each bus master or microprocessor requires an arbiter for the interface to the shared bus.
  - which Intel calls the Multibus
  - and IBM calls the Micro Channel
- Processors connected in this kind of system are often called parallel or distributed processors because they can execute software and perform tasks in parallel.



The ISA Bus

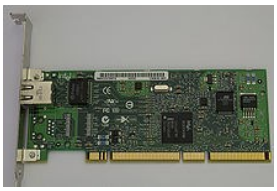
- Industry Standard Architecture
- Originally 8-bit. Later evolved to 16-bit.
- Present in early PCs up to Pentium III
- Still present in some industrial applications

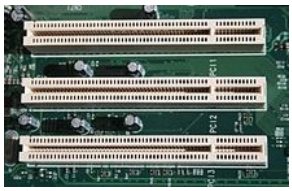
Back of Computer

Pin #	Solder Side	Component Side
1	GND	IO CHK
2	RESET	D7
3	+5V	D6
4	IRQ9	D5
5	–5V	D4
6	DRQ2	D3
7	–12V	D2
8	OWS	D1
9	+12V	D0
10	GND	IO RDY
11	MEMW	AEN
12	MEMR	A19
13	IOW	A18
14	IOR	A17
15	DACK3	A16
16	DRQ3	A15
17	DACK1	A14
18	DRQ1	A13
19	DACK0	A12
20	CLOCK	A11
21	IRQ7	A10
22	IRQ6	A9
23	IRQ5	A8
24	IRQ4	A7
25	IRQ3	A6
26	DACK2	A5
27	T/C	A4
28	ALE	A3
29	+5V	A2
30	OSC	A1
31	GND	A0

The Peripheral Component Interconnect (PCI) Bus

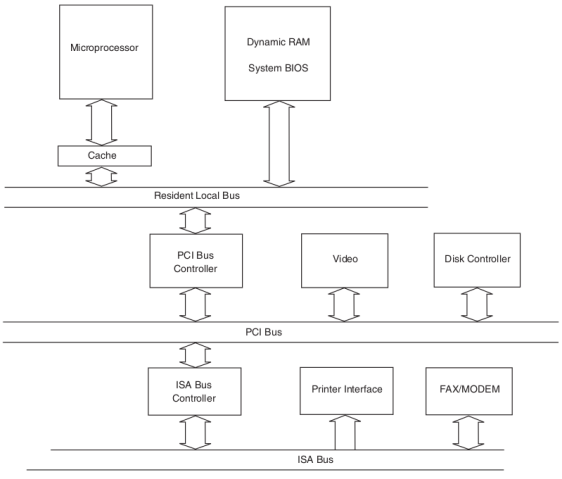
- Plug-and-play characteristics
  - contains a series of registers, located in a small ROM on the PCI interface
- First version PCI 1.0 (1992): 32-bit 33 MHz
- Ability to function with a 64-bit data bus





Pin	Side B	Side A	Comments
1	−12 V	TRST#	
2	TCK	+12 V	
3	Ground	TMS	
4	TDO	TDI	
5	+5 V	+5 V	
6	+5 V	INTA#	
7	INTB#	INTC#	
8	INTD#	+5 V	
9	PRSNT1#	Reserved	
10	Reserved	I/OPWR	
11	PRSNT2#	Reserved	
12	Ground	Ground	
13	Ground	Ground	
14	Reserved	3.3 V aux	
15	Ground	RST#	
16	CLK	I/OPWR	
17	Ground	GNT#	
18	REQ#	Ground	
19	I/OPWR	PME#	
20	AD[31]	AD[30]	
21	AD[29]	+3.3 V	
22	Ground	AD[28]	
23	AD[27]	AD[26]	
24	AD[25]	Ground	
25	+3.3 V	AD[24]	
26	C/BE[3]#	IDSEL	
27	AD[23]	+3.3 V	
28	Ground	AD[22]	
29	AD[21]	AD[20]	
30	AD[19]	Ground	
31	+3.3 V	AD[18]	
32	AD[17]	AD[16]	
33	C/BE[2]#	+3.3 V	
34	Ground	FRAME#	
35	IRDY#	Ground	
36	+3.3 V	TRDY#	
37	DEVSEL#	Ground	
38	PCIXCAP	Ground	STOP#
39	LOCK#	+3.3 V	
40	PERR#	SMBCLK	SDONE
41	+3.3 V	SMBDAT	SBO#
42	SERR#	Ground	
43	+3.3 V	PAR	
44	C/BE[1]#	AD[15]	
45	AD[14]	+3.3 V	
46	Ground	AD[13]	
47	AD[12]	AD[11]	
48	AD[10]	Ground	
49	M66EN	Ground	AD[09]
50	Ground	Ground	
51	Ground	Ground	
52	AD[08]	C/BE[0]#	
53	AD[07]	+3.3 V	
54	+3.3 V	AD[06]	
55	AD[05]	AD[04]	
56	AD[03]	Ground	
57	Ground	AD[02]	
58	AD[01]	AD[00]	
59	I/OPWR	I/OPWR	
60	ACK64#	REQ64#	
61	+5 V	+5 V	
62	+5 V	+5 V	

A PC with PCI Bus

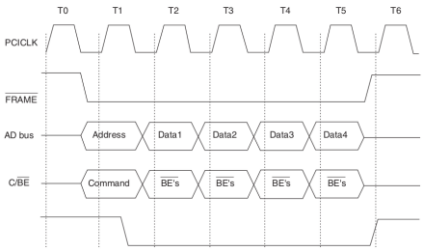


PCI Bus Operation

Bus Commands

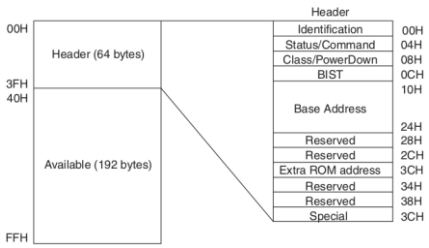
C/$\overline{BE3}$–C/$\overline{BE0}$	*Command*
0000	INTA sequence
0001	Special cycle
0010	I/O read cycle
0011	I/O write cycle
0100–1001	Reserved
1010	Configuration read
1011	Configuration write
1100	Memory multiple access
1101	Dual addressing cycle
1110	Line memory access
1111	Memory write with invalidation

Burst mode Timing



PCI Configuration Space

Configuration Memory



Class Codes

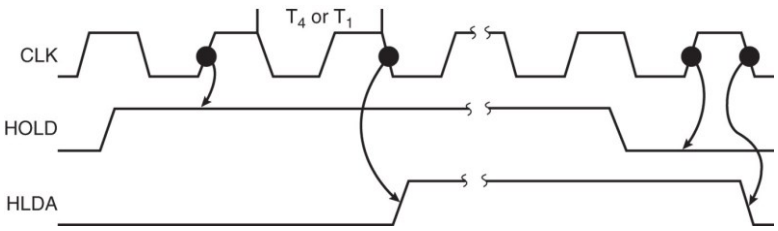
Class Code	*Function*
0000H	Older non-VGA device (not PnP)
0001H	Older VGA device (not PnP)
0100H	SCSI controller
0101H	IDE controller
0102H	Floppy disk controller
0103H	IPI controller
0180H	Other hard/floppy controller
0200H	Ethernet controller
0201H	Token ring controller
0202H	FDDI
0280H	Other network controller
0300H	VGA controller
0301H	XGA controller
0380H	Other video controller
0400H	Video multimedia
0480H	Other multimedia controller
0500H	RAM controller
0580H	Other memory bridge controller
0600H	Host bridge
0601H	ISA bridge
0602H	EISA bridge
0603H	MCA bridge
0604H	PCI–PCI bridge
0605H	PCMIA bridge
0680H	Other bridge
0700H–FFFEH	Reserved
FFFFH	Not installed

Recent Interfacing Standards

- Recent peripheral buses are serial
  - PCI Express
  - USB
  - Thunderbolt
- Achieve higher transfer rate

BASIC DMA OPERATION

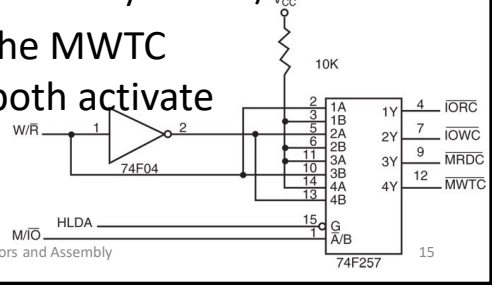
- Two control signals are used to request and acknowledge a DMA transfer
- HOLD input used to request a DMA action
  - Processor stops executing SW and enters hold cycles
  - Has higher priority than INTR or NMI
- HLDA output acknowledges the DMA action
  - Indicates processor has put its busses in Hi-Z



DMA Signals

- Memory and IO controlled simultaneously
  - That's why they have separate control signals
- A DMA read causes the MRDC and IOWC signals to activate simultaneously.
  - transferring data from memory to the I/O device
- A DMA write causes the MWTC and IORC signals to both activate

74F257:
quad 2-line to 1-line multiplexer



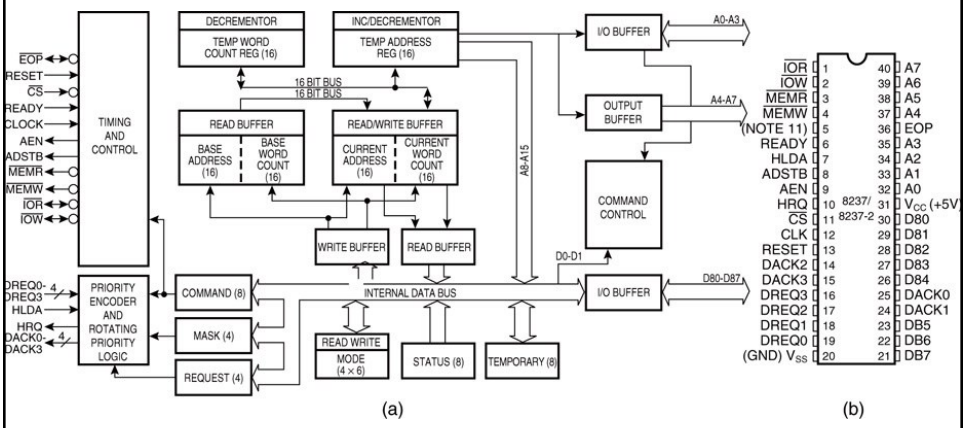
DMA Speed

- Data transfer speed is determined by speed of the memory device or a DMA controller.
  - Mem speed: 50 ns -> Max DMA speed: 20 MB/s
  - But if DMA Freq 15MHz -> slows the system
- The switch to serial data transfers in modern systems has replaced DMAs.
  - The serial PCI Express bus transfers data at rates exceeding DMA transfers.
  - SATA: 300 Mbps
  - On-board PCI Express: 20 Gbps

THE 8237 DMA CONTROLLER

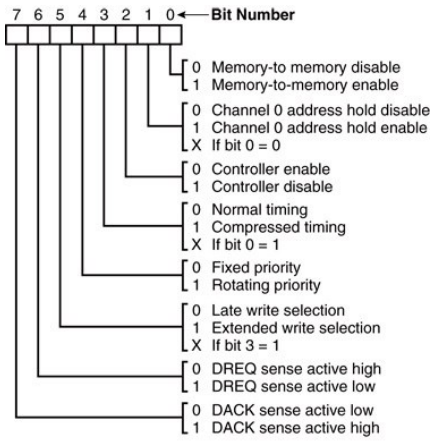
- A four-channel device compatible with 8086/8088, adequate for small systems.
- Capable of DMA transfers at rates up to 1.6M bytes per second.
- Each channel is capable of addressing a full 64K-byte section of memory

THE 8237 DMA CONTROLLER

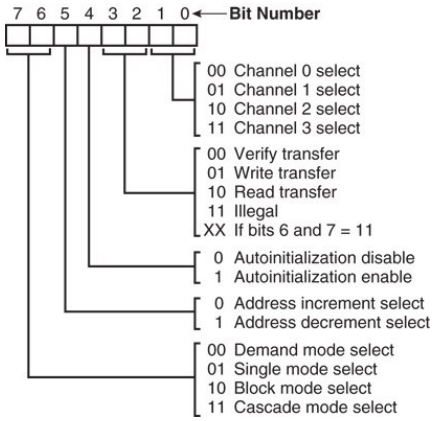


8237 Registers I

Command Register

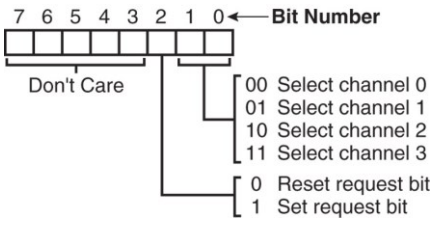


Mode Registers
(One for each channel)

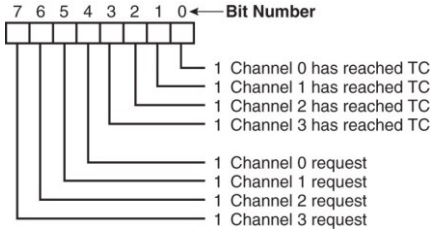


8237 Registers II

Bus Request Register

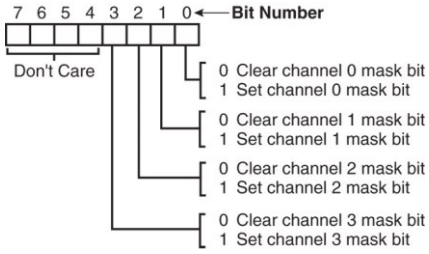


Status Register

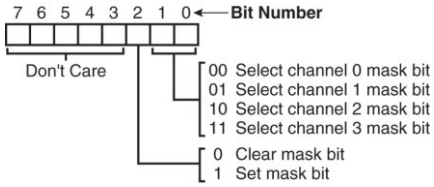


8237 Registers III

Mask Register



Mask Register Set/Reset



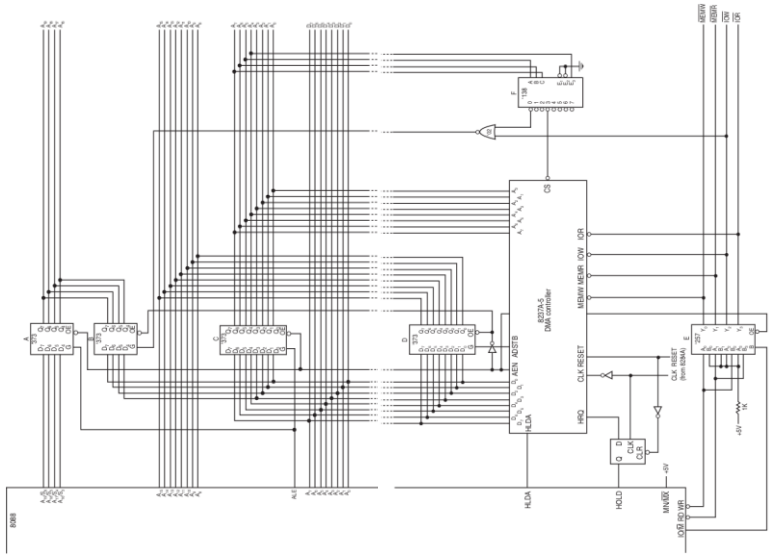
Software Commands

Signals						Operation
A3	A2	A1	A0	$\overline{IOR}$	$\overline{IOW}$	
1	0	0	0	0	1	Read Status Register
1	0	0	0	1	0	Write Command Register
1	0	0	1	0	1	Illegal
1	0	0	1	1	0	Write Request Register
1	0	1	0	0	1	Illegal
1	0	1	0	1	0	Write Single Mask Register Bit
1	0	1	1	0	1	Illegal
1	0	1	1	1	0	Write Mode Register
1	1	0	0	0	1	Illegal
1	1	0	0	1	0	Clear Byte Pointer Flip/Flop
1	1	0	1	0	1	Read Temporary Register
1	1	0	1	1	0	Master Clear
1	1	1	0	0	1	Illegal
1	1	1	0	1	0	Clear Mask Register
1	1	1	1	0	1	Illegal
1	1	1	1	1	0	Write All Mask Register Bits

Programming the Address and Count Registers

Channel	Register	Operation	Signals							Internal Flip-Flop	Data Bus DB0-DB7
			$\overline{CS}$	$\overline{IOR}$	$\overline{IOW}$	A3	A2	A1	A0		
0	Base and Current Address	Write	0	1	0	0	0	0	0	0	A0-A7
			0	1	0	0	0	0	0	1	A8-A15
	Current Address	Read	0	0	1	0	0	0	0	0	A0-A7
			0	0	1	0	0	0	0	1	A8-A15
	Base and Current Word Count	Write	0	1	0	0	0	0	1	0	W0-W7
			0	1	0	0	0	0	1	1	W8-W15
	Current Word Count	Read	0	0	1	0	0	0	1	0	W0-W7
			0	0	1	0	0	0	1	1	W8-W15
1	Base and Current Address	Write	0	1	0	0	0	1	0	0	A0-A7
			0	1	0	0	0	1	0	1	A8-A15
	Current Address	Read	0	0	1	0	0	1	0	0	A0-A7
			0	0	1	0	0	1	0	1	A8-A15
	Base and Current Word Count	Write	0	1	0	0	0	1	1	0	W0-W7
			0	1	0	0	0	1	1	1	W8-W15
	Current Word Count	Read	0	0	1	0	0	1	1	0	W0-W7
			0	0	1	0	0	1	1	1	W8-W15
2	Base and Current Address	Write	0	1	0	0	1	0	0	0	A0-A7
			0	1	0	0	1	0	0	1	A8-A15
	Current Address	Read	0	0	1	0	1	0	0	0	A0-A7
			0	0	1	0	1	0	0	1	A8-A15
	Base and Current Word Count	Write	0	1	0	0	1	0	1	0	W0-W7
			0	1	0	0	1	0	1	1	W8-W15
	Current Word Count	Read	0	0	1	0	1	0	1	0	W0-W7
			0	0	1	0	1	0	1	1	W8-W15
3	Base and Current Address	Write	0	1	0	0	1	1	0	0	A0-A7
			0	1	0	0	1	1	0	1	A8-A15
	Current Address	Read	0	0	1	0	1	1	0	0	A0-A7
			0	0	1	0	1	1	0	1	A8-A15
	Base and Current Word Count	Write	0	1	0	0	1	1	1	0	W0-W7
			0	1	0	0	1	1	1	1	W8-W15
	Current Word Count	Read	0	0	1	0	1	1	1	0	W0-W7
			0	0	1	0	1	1	1	1	W8-W15

Connection to 8088



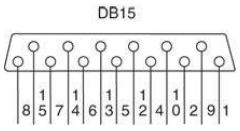
Example:
Clear the DOS Mode Video Screen using DMA

```
;A procedure that clears the DOS mode video screen using the DMA
;controller as depicted in Figure 13-12.

;Calling sequence:
;      DI = offset address of area cleared
;      ES = segment address of area cleared
;      CX = number of bytes cleared

LATCHB EQU   10H
CLEARF EQU   7CH
CH0A   EQU   70H
CH1A   EQU   72H
CH1C   EQU   73H
MODE   EQU   7BH
CMMD   EQU   78H
MASKS  EQU   7FH
REQ    EQU   79H
STATUS EQU   78H
ZERO   EQU   0

CLEAR  PROC  NEAR USES AX

   MOV   AX,ES             ;program latch B
   MOV   AL,AH
   SHR   AL,4
   OUT   LATCHB,AL
   OUT   CLEARF,AL         ;clear F/L

   MOV   AL,ZERO           ;save zero in first byte
   MOV   ES:[DI],AL

   MOV   AX,ES             ;program source address
   SHL   AX,4
   ADD   AX,SI
   OUT   CH0A,AL
   MOV   AL,AH
   OUT   CH0A

   MOV   AX,ES             ;program destination address
   SHL   AX,4
   ADD   AX,DI
   OUT   CH1A,AL
   MOV   AL,AH
   OUT   CH1A,AL

   MOV   AX,CX             ;program count
   DEC   AX
   OUT   CH1C,AL
   MOV   AL,AH
   OUT   CH1C,AL

   MOV   AL,88H            ;program mode
   OUT   MODE,AL
   MOV   AL,85H
   OUT   MODE,AL

   MOV   AL,03H            ;enable block hold transfer
   OUT   CMMD,AL

   MOV   AL,0EH            ;enable channel 0
   OUT   MASKS,AL

   MOV   AL,4              ;start DMA
   OUT   REQ,AL

   .REPEAT
      IN   AL,STATUS
   .UNTIL AL &  1
   RET

CLEAR     ENDP
```

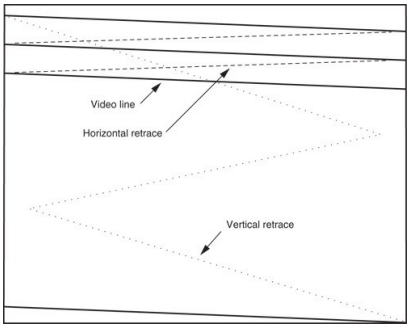

DISK MEMORY SYSTEMS

- Disk memory is used to store long-term data.
- Many types of disk storage systems are available and they use magnetic media.
- Optical disk memory
  - CD-ROM, DVD
- Flash drives

Video Displays: RGB Analog Monitor

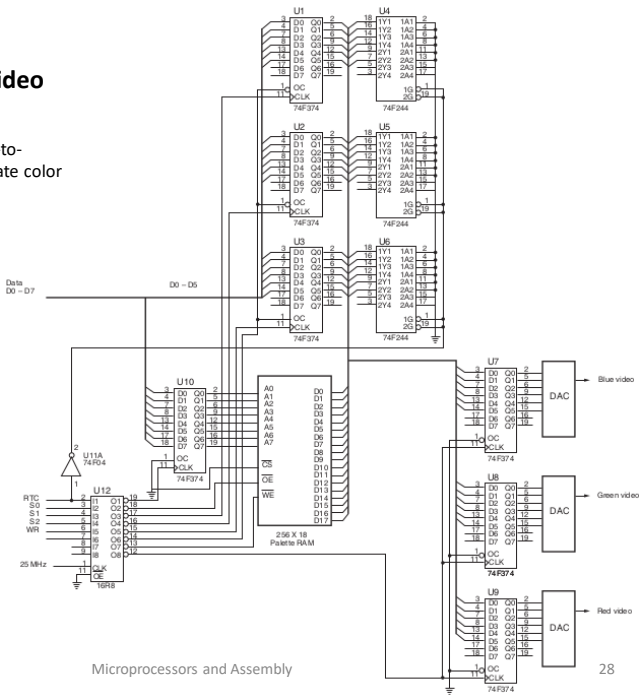


Pin	Function
1	Red video
2	Green video (monochrome video)
3	Blue video
4	Ground
5	Ground
6	Red ground
7	Green ground (monochrome ground)
8	Blue ground
9	Blocked as a key
10	Ground
11	Color detect (ground on a color monitor)
12	Monochrome detect (ground on a monochrome monitor)
13	Horizontal retrace
14	Vertical retrace
15	Ground

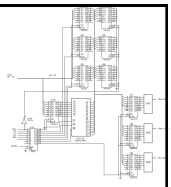


Generation of VGA video signals

- analog displays use digital-to-analog converter to generate color video voltage

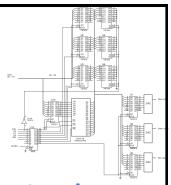


VGA Signals



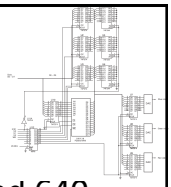
- A high-speed palette SRAM is used to store 256 different 18-bit codes representing hues.
- To select any of 256 colors, an 8-bit code stored in the computer's video display RAM is used to specify color of a picture element.
  - newer systems use larger palette SRAM
- If color codes must be changed, it is done during retrace when RTC is a logic 1.

VGA Signals



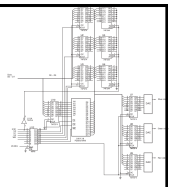
- Retrace occurs 70.1 times per second vertical and 31,500 times per second horizontal direction for a 640 × 480 display.
  - used to move the electron beam to the upper left corner for vertical retrace and the left margin of the screen for horizontal retrace
- The resolution of the display determines the memory required for the video interface card.
  - 640 × 480 bytes of memory (307,200) are required to store all of the pixels for the display

VGA Signals



- A 640 × 480 display has 480 video raster lines and 640 pixels per line.
  - a **raster line** is the horizontal line of video information that is displayed on the monitor
  - a pixel (picture element) is the smallest subdivision of this horizontal line
- In order to generate 640 pixels across one line, it takes 40 ns × 640, or 25.6 μs.
- A horizontal time of 31,500 Hz allows a horizontal line time of 1/31,500, or 31.746 μs.
  - the difference between these two times is the retrace time allowed to the monitor

VGA Signals



- In the case of a VGA display (a 640 × 400 display), this is 449.358 lines.
  - 400 lines are used to display information
  - the rest are lost during the retrace
- Because 49.358 lines are lost, retrace time is 49.358 × 31.766 μs, or 1568 μs.
- During this time color palette SRAM is changed or the display memory is updated.