Unit II
Bus Cycles and System Architecture

Prof. Sai Takawale
Department of Computer Engineering
Trinity college of Engineering and Research,Pune

Syllabus

Part A: Initialization :

1. Processor State after reset
2. Functional Pin Diagram
3. Functionality of various pins
4. I/O Organization
5. Memory Organization
6. Basic memory read and write cycles with timing diagram

Syllabus

Part B: System Architecture :

1. System Register :
   a. System Flag
   b. Memory Management register
   c. Control Register
   d. Debug register
   e. Test Register
1. System Instruction

Initialization

After a signal on the RESET pin, certain registers of the 80386 are set to predefined values.

These values are adequate to enable execution of a bootstrap program, but additional initialization must be performed by software before all the features of the processor can be utilized.

Processor State after Reset

EAX Register : The contents of EAX depend upon the results of the power-up self test. The self-test may be requested externally by assertion of BUSY# at the end of RESET. The EAX register holds zero if the 80386 passed the test. A nonzero value in EAX after self-test indicates that the particular 80386 unit is faulty. If the self-test is not requested, the contents of EAX after RESET is undefined.

Processor State after Reset

DX Register : DX holds a component identifier and revision number after RESET as shown in Fig. 3.2.1. OH contains 3, which indicates an 80386 component. DL contains a unique identifier of the revision level.

CRO Register : Control register zero (CRO) contains the values shown in Fig. 3.2.2. The ET bit of CRO is set if an 80387 is present in the configuration (according to the state of the ERROR# pin after RESET). If ET is reset, the configuration either contains an 80287 or does not contain a coprocessor. A software test is required to distinguish between these latter two possibilities.

Processor State after Reset

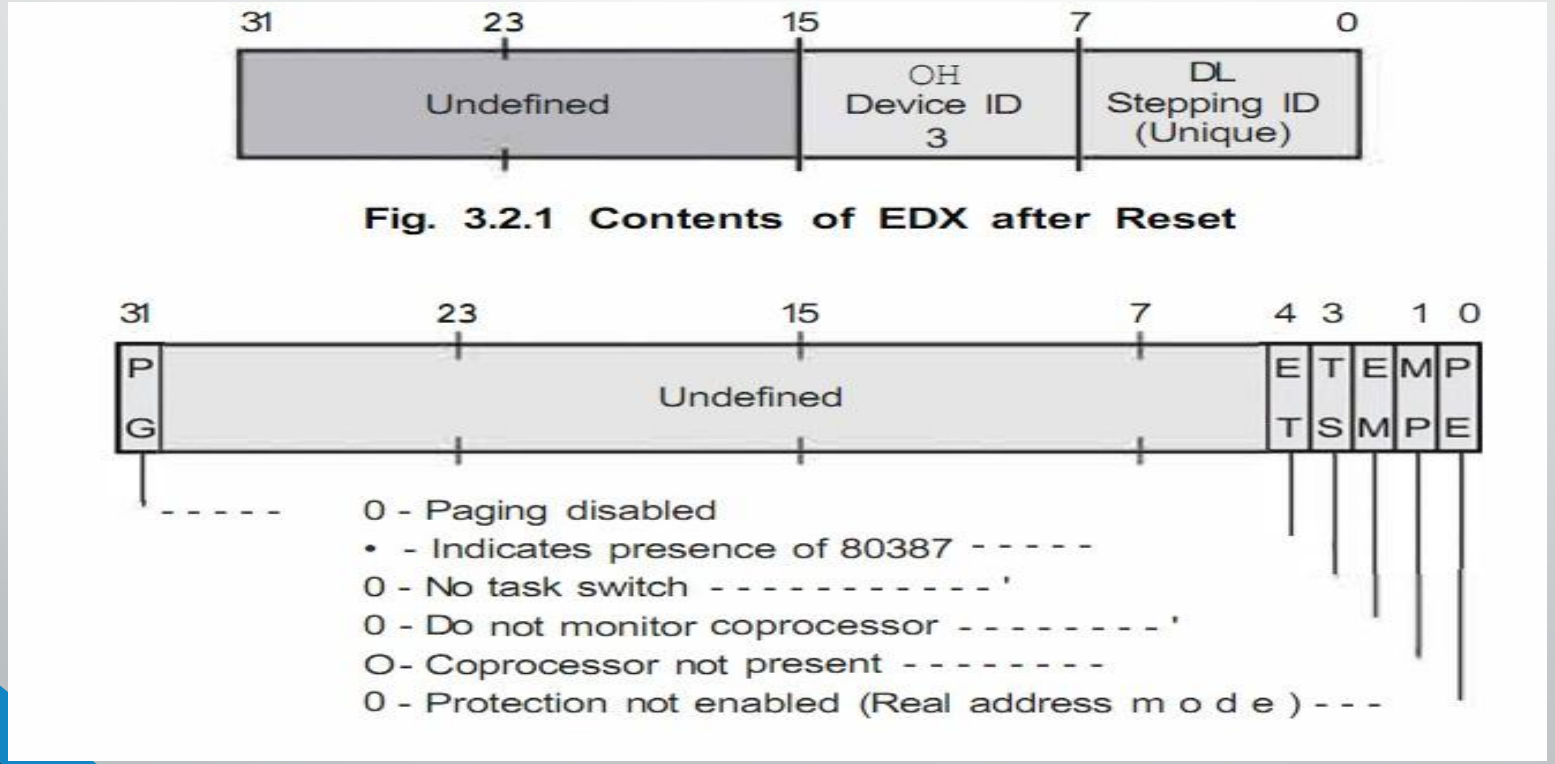


Fig. 3.2.1 Contents of EDX after Reset

Processor State after Reset

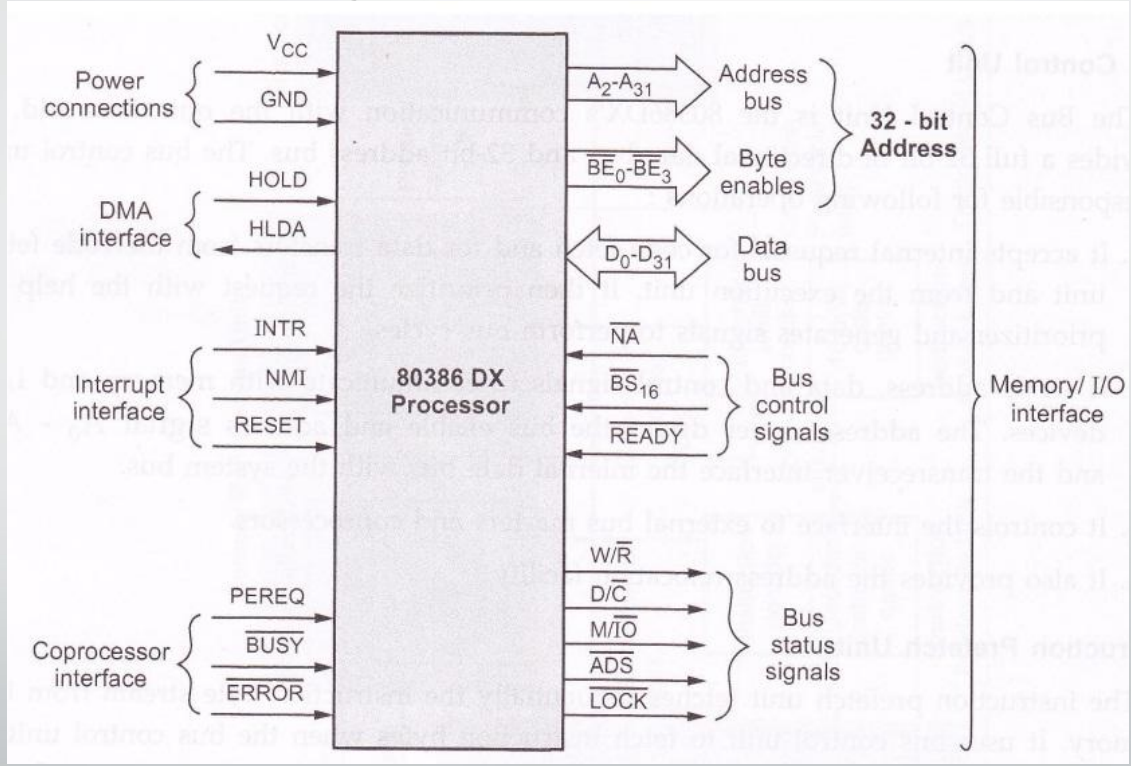
These settings imply that the processor begins in real-address mode with interrupts disabled.

Processor State after Reset

Remaining registers and flags Status :

EFLAGS = 00000002H	SS selector = 0000H
IP = OO00FFF0H	FS selector = 0000H
CS selector = 0000H	GS selector = 0000H
OS selector = 0OO0H	IDTR: base = 0 limit = 03FFH
ES selector = 0000H	**All** registers not mentioned **above are undefined.**

Functional Pin Diagram of 80386



Functionality of various pins

These signals are separated in four major groups :

1. Memory /IO Interface
2. Interrupt Interface
3. DMA Interface
4. Coprocessor Interface

Memory/IO Interface Signals

It includes

1. **Data bus: D_{31}-D_0**
2. **Separate address bus: A_{31}-A_2 and BE_3-BE_0**
3. **Five bus status signals: Address status (ADS) , Write/Read (W/R), Memory/I/o (M/Io) , Data/Control (D/C) , LOCK**
4. **Three bus control signals: Bus Control Signals , READY , Next Address Request (NA) , Bus Size 16 (BS16)**

Bus Status Signals

M/IO	*DIC*	W/R	Type of Bus Cycle
0	0	0	Interrupt acknowledge
0	0	1	Idle
0	1	0	I/0 data read
0	1	1	1/0 data write
1	0	0	Memory code read
1	0	1	Halt/shutdown
1	1	0	Memory data read
1	1	1	Memory data write

Bus Status Signals

Address status (ADS) : A low on this pin indicates that the valid address is present on the address bus.

Write/Read (W/R) : This signal decides the specific type of memory or I/O operation that will occur during a bus cycle.

Memory/I/o (M/Io) : This signal identifies whether the current bus cycle is memory cycle or I/o cycle.

Data/Control (D/C) : This signal identifies whether the current bus cycle is data or control cycle.

Bus Status Signals

LOCK : This signal is used in multiprocessor systems. In multiprocessor systems resources are shared, such as global memory.
The LOCK is signal is used to ensure that

The 80386DX has uninterrupted control of the system bus and the shared resource. By making LOCK output 0,

The 80386DX can lock up the shared resource for the other masters in the system.

Bus Control Signals

READY : It is used primarily to synchronize slower peripherals with the microprocessor.

This signal is produced by the microcomputer's memory or I/O subsystem.

When READY signal is logic 0, slow memory or I/O devices tell 80386DX that they are ready for next data transfer.

If ready is logic 1 then processor enters wait state since logic 1 on ready pin indicates that, the data transfer of current cycle is not yet completed.

Bus Control Signals

Next Address Request (NA) : The external bus control logic control this signal.

It activates pipelining by making next address request input low.

Pipelining increases the address to data access time.

By increasing the address to data access time, same level of performance can be obtained with slower, memory devices.

Bus Control Signals

Bus Size 16 (BS16) : This signal activates 16-bit data bus operation;

data is transferred on the low-order 16-bits of the data bus,

and an extra cycle is provided for transfers of more than 16-bits.

Interrupt Interface Signals

INTR : The INTR input of the 80386 allows external devices to interrupt 80386 program execution.

This input is sampled at the beginning of each instruction cycle. To ensure recognition of interrupt by the 80386,

The INTR input must be held high until the 80386 acknowledges the interrupt by performing the interrupt acknowledge cycles.

Thus it must be high at least eight CLK2 periods prior to the instruction to guarantee recognition as a valid interrupt.

This specific requirement reduces the false triggering.

Interrupt Interface Signals

Nonmaskable Interrupt (NMI) : As name indicates this interrupt input is nonmaskable.

This input is edge-triggered. A valid interrupt on this pin causes 80386 to execute interrupt service routine.

The 80386 will not service subsequent NMI requests until the current request has been serviced.

Interrupt Interface Signals

Reset : Reset input forces 80386 to go into the reset state.

This is an active high signal. When this signal in high it resets system resources, such as 1/0 ports, and the interrupt flag.

After reset 80386 starts execution of program from memory address FFFFFFF0H in real mode.

1111

DMA Interface

HOLD and HLDA : These pins are used to interface OMA controller.
The OMA controller can request for bus access by asserting HOLD pin and 80386DX tells the OMA controller that the buses are available by asserting HLDA signal.

The 80386DX activates its HLDA signal after completion of current bus cycle and it

enters in HOLD state. In HOLD state, its local bus signals are in high impedance state.

Coprocessor Interface Signals

BUSY : BUSY and ERROR are status signals from the coprocessor.

BUSY signal is used to tell 80386 that the coprocessor is executing a numeric instruction.

Thus when BUSY is logic 0, 80386 does not request the numeric coprocessor to perform another calculation until BUSY returns to logic 1.

Coprocessor Interface Signals

ERROR : ERROR signal is used to indicate occurrence of error in the calculation.

If an error occurs in a calculation performed by the numeric coprocessor,

it informs 80386 that error has occurred by activating ERROR signal.

Coprocessor Interface Signals

PEREQ : The coprocessor cannot transfer data over the data bus by itself.

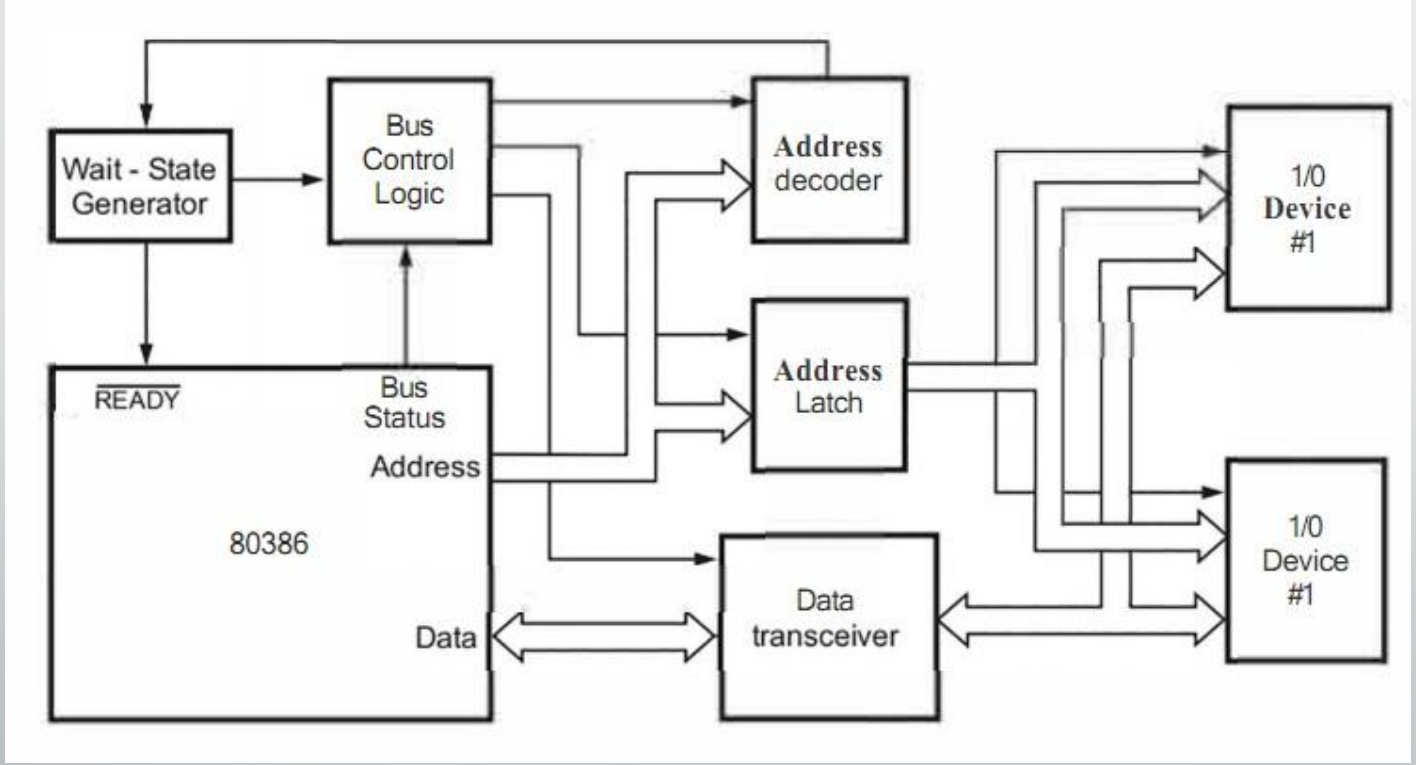
Whenever the coprocessor needs to read or write data from memory, it signals 80386 to initiates data transfer.

The coprocessor does this by making PEREQ signal high.

I/O Organization

The 80386 supports 8-bit, 16-bit, and 32 bit 1/0 devices that can be mapped into either the 64-kilobyte
I/o address space or the 4-gigabyte physical memory address space.

I/O Organization



I/O Organization

Input/Output devices can be interfaced with 80386 systems in two ways.

1. I/O mapped I/O

2. Memory mapped I/O

I/O mapped I/O

In I/o mapped I/o, the 1/o devices are treated separate from memory.

The 80386 supports software and hardware features for separate memory and I/o

address spaces. Fig. 3.4.2 shows the memory and I/o spaces in real mode.

The 80386 has four special instructions IN, INS, OUT, and OUTS to transfer data

through input/output ports in I/o mapped I/o system. M/10 signal is always low when 80386 is executing these instructions. So M/IO

signal is used to generate separate addresses for Input/Output.

Memory mapped I/O

In memory mapped I/o, I/o device is placed in the memory address space of the microcomputer system.

I/O device is connected as if it is a memory location. For this reason, the method is known as memory mapped I/O.

In memory-mapped 1/0, some of the memory address spaces are dedicated to the 1/0 system.

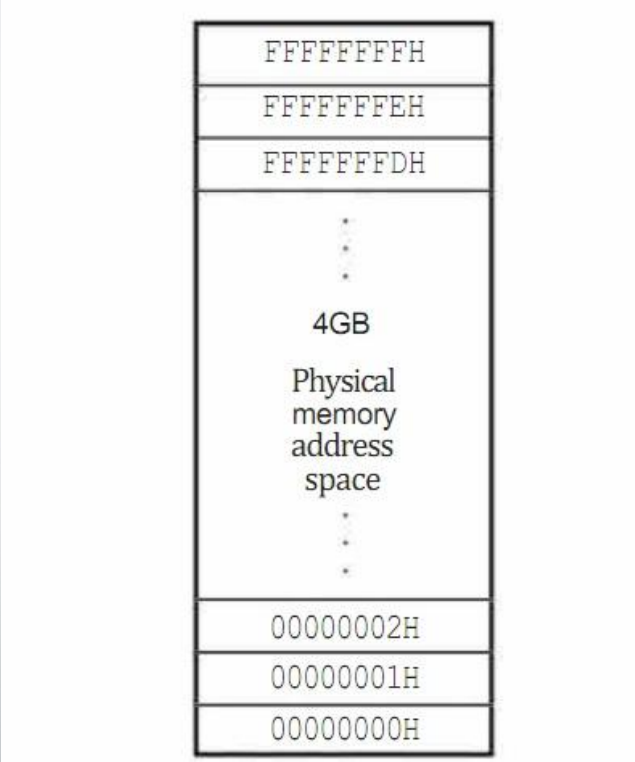
Memory Organization (Memory Banks)

The 80386DX has 32-bit address bus so it can access upto 4G-bytes (2 32)

of memory locations.

• Fig. 3.5.1 shows the physical address space. From the software point of view this memory is organized over the address range from oooooooH through FFFFFFFFH and 80386DX can access data in this memory as byte, word or double words.

Memory Organization (Memory Banks)



Memory Organization (Memory Banks)

The words are accessed from two consecutive memory locations whereas double

words are accessed from four consecutive memory locations.

• To implement this entire memory is divided into four independent byte-wide

memory banks: Banko-Bank3. Each bank is lG-byte in size.

Memory Organization (Memory Banks)

As shown in figure bank 0, bank 1, bank 2 and bank 3 are selected using byte

enable signals BE0, BEl, BE2 and BE3 signals, respectively.

Address lines A31-A2 are connected in parallel to all memory banks which make

it possible to access lG-byte of memory. But the 32-bit data bus is distributed over

four memory banks, bit each.

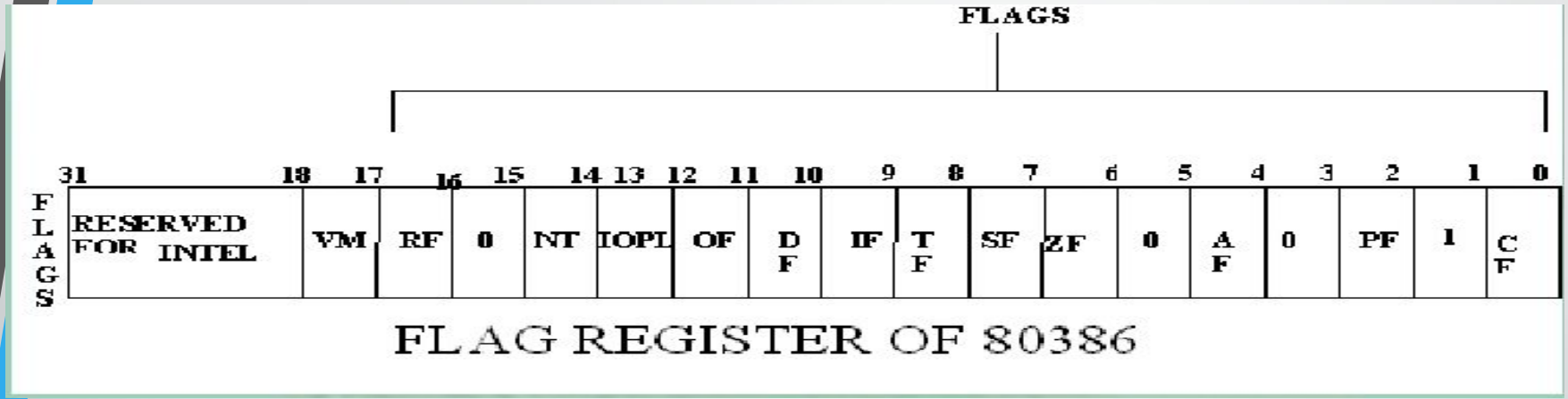
Basic Memory Read and Writes Cycles with Timing Diagram

1. Non-pipelined write cycle : Click
2. Non pipelined read cycle : Click
3. Pipelined Read-Write Cycle: Click

Part B: System Architecture :

1. System Register :
   a. System Flag
   b. Memory Management register
   c. Control Register
   d. Debug register
   e. Test Register
1. System Instruction

Flag Register



FLAG REGISTER OF 80386

Flag Register of 80386: The Flag register of 80386 is a 32 bit register. Out of the 32 bits, Intel has reserved bits D18 to D31, D5 and D3, while D1 is always set at 1.Two extra new flags are added to the 80286 flag to derive the flag register of 80386. They are VM and RF

System Address Registers

The 386 supports four types of descriptor table:

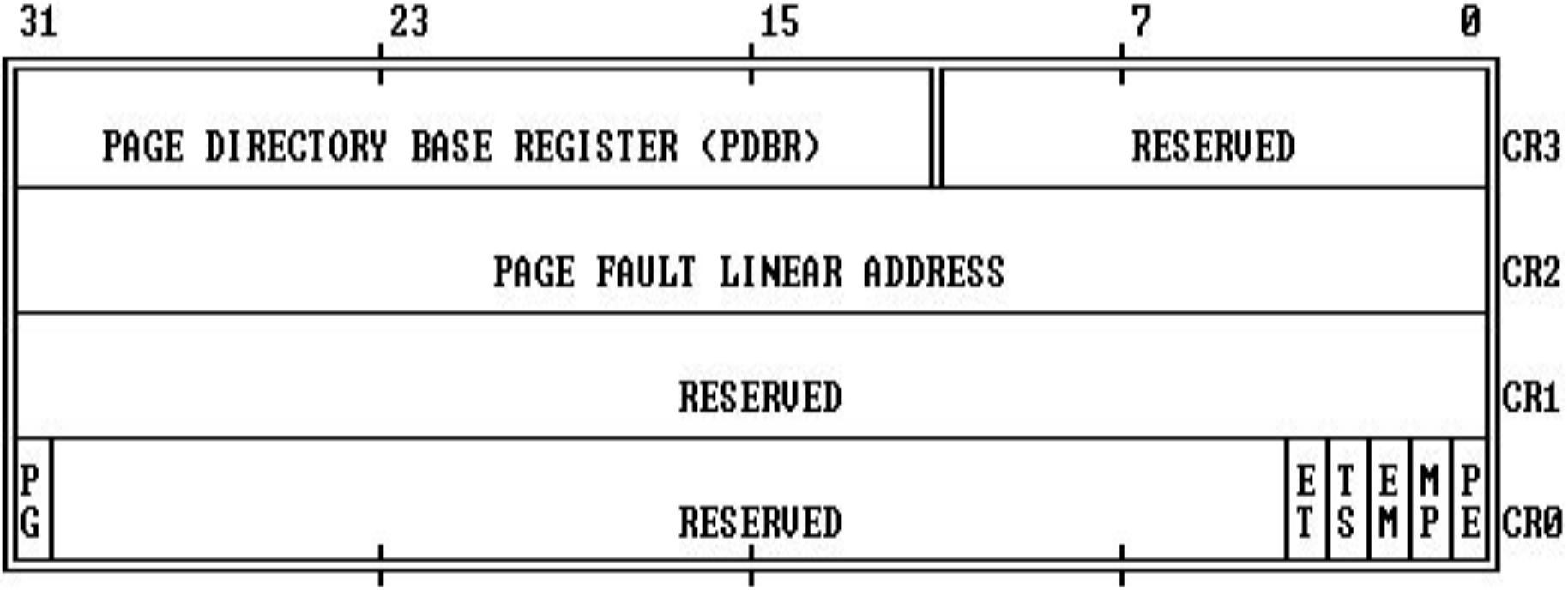
- Global descriptor table (GDT),
- Local descriptor table (LDT),
- Interrupt descriptor table (IDT), and
- Task state segment descriptor (TSS).

Four special registers are defined to hold the base address of these tables

- Global descriptor table Register (**GDTR),**
- Local descriptor table Register (**LDTR)**

Control Register(32- bit)

Figure 4-2. Control Registers



Control Register 0 (CR0)

CR0 contains system control flags, which control or indicate conditions that apply to the system as a whole, not to an individual task (Also know as MSW).

- **PE (Protection Enable bit 0)**
  - Setting PE causes the processor to begin executing in protected mode. Resetting PE returns to real-address mode (PE=1 System in protected mode else in real mode)
- **MP (Math Present, bit 1)**
  - The MP (monitor coprocessor) bit indicates whether a coprocessor is actually attached
- **EM (Emulation, bit 2)**
  - EM=1 cause type 7 interrupt for each ESC instruction.

Control Register 0 (CR0)

- **TS (Task Switch, bit 3)**
  - Indicate task has been switched to 80387 (TS=1 numeric coprocessor cause type7 interrupt).
- **ET (Extension Type, bit 4)**
  - ET indicates the type of coprocessor present in the system (80287 or 80387) . (ET=0 select 80287; ET=1 select 80387)
- **PG(Paging , bit 31)**
  - PG indicates whether the processor uses page tables to translate linear

Control Register

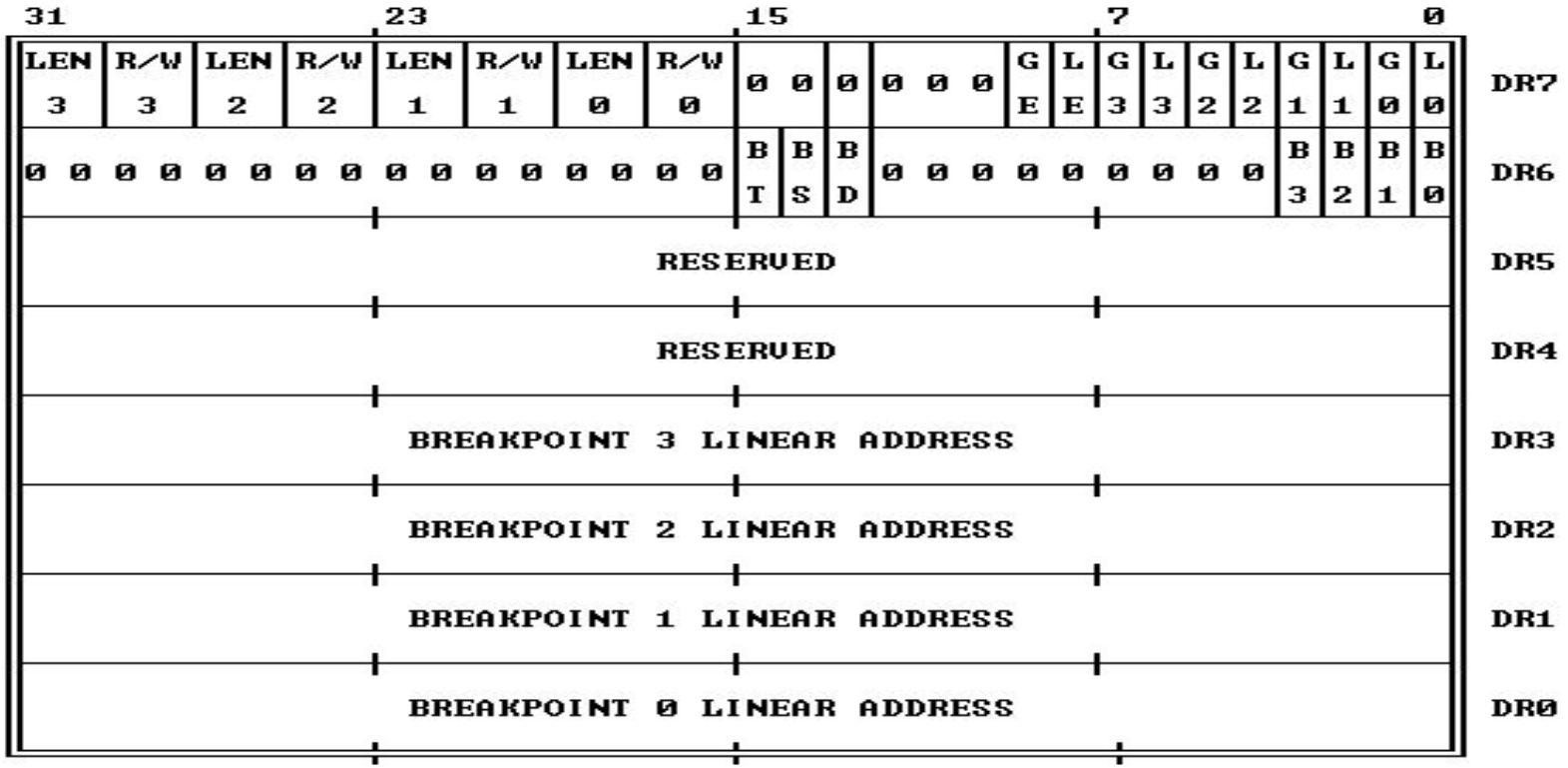
- **Control Register 1 (CR1).**
  - reserved by Intel
- **Control Register 2 (CR2)**
  - Read only register. The 80386 itself writes the last 32 bit linear address of page fault routine in this register. When page fault occurs , porcessor generates exception 14 (page fault) .
  - This address is important for writing page fault routine.
- **Control Register 3 (CR3)**
  - Control register 3 hold the base address of page directory

Debug Register

- Intel has provide a set of 8 debug registers for hardware debugging. Out of these eight registers DR0 to DR7, two registers DR4 and DR5 are Intel reserved.
- The initial four registers DR0 to DR3 store four program controllable breakpoint addresses, while DR6 and DR7 respectively hold breakpoint status and breakpoint control information.
- Breakpoint address may locate an instruction or datum.

Debug Register

Figure 12-1. Debug Registers



NOTE

0 MEANS INTEL RESERVED. DO NOT DEFINE.

Debug Address Register (DR0-DR3)

- Each of these registers contains the linear address associated with one of four breakpoint conditions. Each breakpoint condition is further defined by bits in DR7.
- The debug address registers are effective whether or not paging is enabled. The addresses in these registers are linear addresses.

Debug Status Register(DR7)

- The debug status register permits the debugger to determine which debug conditions have occurred.
- For each address in registers DR0-DR3, the corresponding fields R/W0 through R/W3 specify the type of action that should cause a breakpoint. The processor interprets these bits as follows:

  00 -- Break on instruction execution only

  01 -- Break on data writes only

  10 -- undefined

Debug Status Register(DR7)

- The LE and GE bits control the "exact data breakpoint match" feature of the processor. If either LE or GE is set, the processor slows execution so that data breakpoints are reported on the instruction that causes them. It is recommended that one of these bits be set whenever data breakpoints are armed. The processor clears LE at a task switch but does not clear GE.

Test Register

Among the eight test registers (TRo-TR7) , only two test registers (TR) are currently defined.

These registers are used to check Translation Lookaside Buffer (TLB) of the paging Unit

Test Register

Test Register 6 : This is the TLB testing command register. By writing into this register, it is possible to either initiate a write directly into the 80386's TLB or to perform TLB lookups.

Test Register 7 : This register is the data testing register of the TLB.

System Instruction

Systems instructions deal with such functions as

Verification of pointer parameters :

ARPL - Adjust RPL

LAR - Load Access Rights LSL Load

Segment Limit VERR - Verify for Reading VERW - Verify for Writing

System Instruction

Addressing descriptor tables :

LLDT - Load LDT Register SLOT - Store LDT Register LGDT - Load GDT Register SGDT - Store GDT Register

Multitasking :

LTR - Load Task Register STR - Store Task Register

System Instruction

Coprocessing and multiprocessing

CL TS Clear

Task-Switched Flag ESC Escape

instructions WAIT - Wait until coprocessor not Busy LOCK - Assert Bus-Lock Signal

System Instruction

Input and output:

IN Input

OUT

Output

INS - Input String OUTS Output

String

Interrupt control :

CLI - Clear Interrupt-Enable Flag STI Set

Interrupt-Enable

Flag LIDT Load

IDT Register STDT - Store IDT Register

System Instruction

Debugging :

MOV - Move to and from debug registers

TLB testing :

MOV - Move to and from test registers

System Instruction

System control:

SMSW Set

MSW

LMSW

Load

MSW

HLT

-

Halt Processor

System Instruction

The instructions SMSW and LMSW are provided for compatibility with the 80286 processor. 80386 programs access the MSW in CR0 via variants of the MOV instruction. HLT stops the processor until receipt of an INTR or RESET signal.