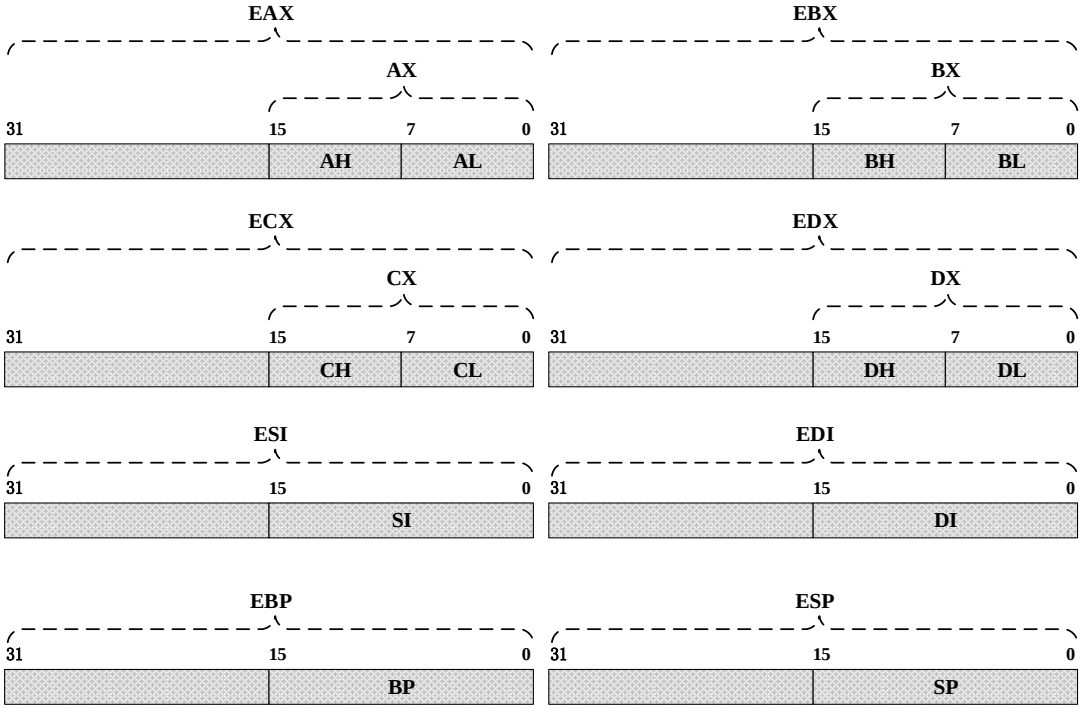
作业 1.1　x86 寄存器详解

一、通用寄存器[1](General-Purpose Registers)

32 位处理器在 16 位处理器的基础上，拓展(Extend)了 8 个通用寄存器(AX、BX、CX、DX、SI、DI、BP、SP)的长度，并将它们命名为 EAX、EBX、ECX、EDX、ESI、EDI、ESP 和 EBP，同时支持 8 位和 16 位操作，用法和 8086 相同。其示意图如下：



虽然这些寄存器都叫通用寄存器，但它们每一个都有自己的特别之处[2]：

1. 数据寄存器

数据寄存器主要用来保存操作数和运算结果等信息，从而节省读取操作数所需占用总线和访问存储器的时间。

- EAX 是累加器(Accumulator Register)，它是很多加法乘法指令的缺省寄存器。
- EBX 是基址寄存器(Base Register)，在内存寻址时用于存放基地址。
- ECX 是计数器(Count Register)，用于记录循环次数以及移位时用 CL 指明移位的位数。
- EDX 是数据寄存器(Data Register)，它可在计算乘除法时用于存放默认操作数，也可用于存放 I/O 的端口地址。

2. 变址寄存器

寄存器 ESI、EDI、SI 和 DI 称为变址寄存器(Index Register)，其中 ESI 和 SI 称为源地址指针寄存器，EDI 和 DI 称为目的地址指针寄存器，它们主要用于存放存储单元在段内的偏移量，它们可作一般的存储器指针使用。在很多字符串操作指令中，DS:ESI 指向源串，而 ES:EDI 指向目标串。

---

[1] 李忠，王晓波，余洁．x86 汇编语言：从实模式到保护模式[M]．北京：电子工业出版社，2012：183-184．

[2] 飘零过客．32 位处理器的寄存器介绍
[DB/OL]．https://blog.csdn.net/xuehuafeiwu123/article/details/76019828，2017-07-24．

3. 地址指针寄存器

32 位 CPU 有 2 个 32 位通用寄存器 EBP 和 ESP。它们主要用于访问堆栈内的存储单元，并且规定：

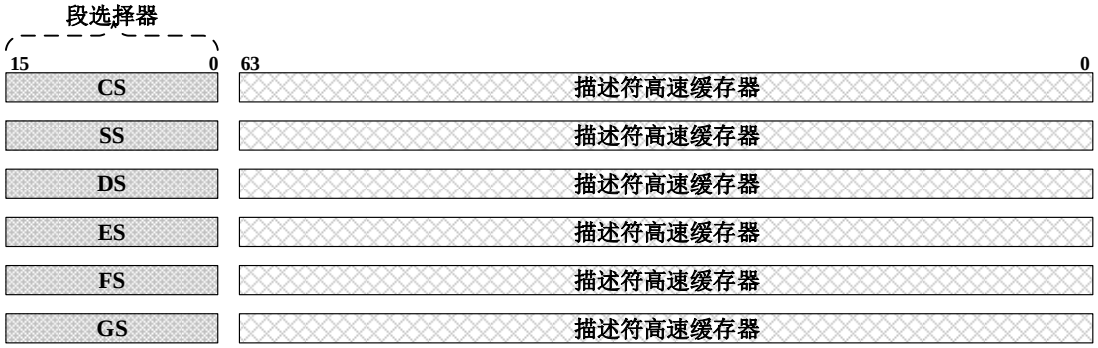
- EBP 是基址指针寄存器(Base Pointer Register)，一般作为当前堆栈的最后单元，用它可直接存取堆栈中的数据；该寄存器可以被作为 BP 或者 EBP 寻址，其缺省段寄存器为 SS。
- ESP 是堆栈指针寄存器(Stack Pointer Register)，专门用作堆栈指针，称之为栈顶指针，在 32 位平台上，ESP 每次减少 4 字节。

二、段寄存器[3](Segment Registers)

除了 8086 中的四个段(CS、DS、ES、SS)外，32 位处理器增加了两个段(FS、GS)，这些段寄存器都是 16 位的，它们的含义如下：

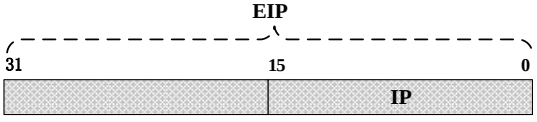
- CS：代码段(Code Segment)
- DS：数据段(Data Segment)
- ES：附加段(Extra Segment)
- SS：堆栈段(Stack Segment)
- FS：标志段(Flag Segment)
- GS：全局段(Global Segment)

在 32 位模式下，传统的段寄存器，如 CS、SS、DS、ES，保存的不再是 16 位段基地址，而是段的选择子，即，用于选择所要访问的段，因此，严格地说，它的新名字叫做段选择器。除了段选择器之外，每个段寄存器还包括一个 64 位的不可见部分，称为描述符高速缓存器，里面有段的基地址和各种访问属性（界限、权限）。这部分内容程序不可访问，由处理器自动使用。示意图如下：



三、指令指针寄存器(Instruction Pointer Register)

指令指针寄存器 EIP 的低 16 位就是 8086 的 IP 寄存器，为了生成 32 位物理地址，处理器需要使用 32 位的指令指针寄存器。示意图如下：



[3] 李忠，王晓波，余洁．x86 汇编语言：从实模式到保护模式[M]．北京：电子工业出版社，2012：185-186．

四、标志寄存器[4][5][6](Program Status and Control Register)

EFLAGS 和 8086 的 16 位标志寄存器相比，增加了 4 个控制位(IOPL、NT、RF、VM)，但这 4 个标志位在实模式下不起作用，它们的位置如下图：

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	ID	VIP	VIF	AC	VM	RF	0	NT	IOPL		OF	DF	IF	TF	SF	ZF	0	AF	0	PF	1	CF

相关的控制/标志位含义如下：

- CF(Carry Flag)：进位标志位
- PF(Parity Flag)：奇偶标志位
- AF(Auxiliary Carry Flag)：辅助进位标志位
- ZF(Zero Flag)：零标志位
- SF(Sign Flag)：符号标志位
- IF(Interrupt Enable Flag)：中断允许标志位，由 CLI、STI 两条指令来控制；设置 IF 使 CPU 可识别外部(可屏蔽)中断请求。复位 IF 则禁止中断。IF 对不可屏蔽外部中断和故障中断的识别没有任何作用。
- DF(Direction Flag)：向量标志位，由 CLD、STD 两条指令来控制。
- OF(Overflow Flag)：溢出标志位。
- IOPL(I/O Privilege Level)：I/O 特权级字段，它的宽度为 2 位，它指定了 I/O 指令的特权级。如果当前的特权级别在数值上小于或等于 IOPL，那么 I/O 指令可以执行。否则，将发生一个保护性故障中断。
- NT(Nested Task Flag)：控制中断返回指令 IRET，它宽度为 1 位。若 NT=0，则用堆栈中保存的值恢复 EFlags、CS 和 EIP 从而实现中断返回；若 NT=1，则通过任务切换实现中断返回。
- TF(Trap Flag)：自陷(追踪)标志位，当 TF 被置为 1 时，CPU 进入单步执行方式，即每执行一条指令，产生一个单步中断请求。这种方式主要用于程序的调试。
- RF(Restart Flag)：重启动标志位，RF 用来控制是否接受调试故障。规定：RF=0 时，表示“接受”调试故障，否则拒绝之。在成功执行完一条指令后，处理机把 RF 置为 0，当接受到一个非调试故障时，处理机就把它置为 1。
- VM(Virtual 8086 Mode)：虚拟 8086 方式标志位，如果 VM 的值为 1，则表示处理机处于虚拟的 8086 方式下的工作状态，否则处理机处于一般保护方式下的工作状态。
- AC(Alignment Check / Access Control)、VIF(Virtual Interrupt Flag)、VIP(Virtual Interrupt Pending)、ID(ID Flag)是 Pentium 处理器增加的标志位。

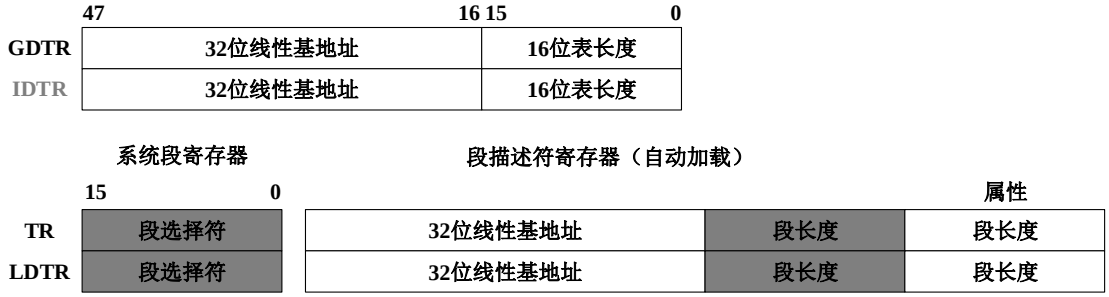
五、系统地址寄存器[7](Memory Management Registers)

80386 有 4 个系统地址寄存器，它们是全局描述符表寄存器 GDTR(Global Descriptor Table Register)、中断描述符表寄存器 IDTR(Interrupt Descriptor Table Register)、局部描述表寄存器 LDTR(Local Descriptor Table Register)和任务寄存器 TR(Task Register)。它们主要用来在保护模式下，管理用于生成线性地址和物理地址的 4 个系统管理描述符表，这些寄存

---

[4] 陈渝，向勇．操作系统实验指导[M]．北京，清华大学出版社，2013：33-34．
[5] qintangtao．指令指针寄存器和标志寄存器
[DB/OL]．http://www.cnblogs.com/qintangtao/p/4161912.html，2014-12-13．
[6] feng.qi．指令指针寄存器和标志寄存器[DB/OL]．https://www.cnblogs.com/feng-qi/articles/3055729.html，2013-05-03．
[7] yyt7529．x86 中内存管理寄存器[DB/OL]．https://blog.csdn.net/yyt7529/article/details/4325980，2009-07-06．

器不直接被程序访问。[8]



GDTR、LDTR、IDTR 和 TR 都是段基址寄存器，这些段中含有分段机制的重要信息表。GDTR、IDTR 和 LDTR 用于寻址存放描述符表的段。TR 用于寻址一个特殊的任务状态段(Task State Segment，TSS)。TSS 中包含着当前执行任务的重要信息。

- 全局描述符表寄存器 GDTR(Global Descriptor Table Register)：48 位寄存器，GDTR 寄存器中用于存放全局描述符表 GDT 的 32 位的线性基地址和 16 位的表限长值。基地址指定 GDT 表中字节 0 在线性地址空间中的地址，表长度指明 GDT 表的字节长度值。指令 LGDT 和 SGDT 分别用于加载和保存 GDTR 寄存器的内容。在机器刚加电或处理器复位后，基地址被默认地设置为 0，而长度值被设置成 0xFFFF。在保护模式初始化过程中必须给 GDTR 加载一个新值。
- 中断描述符表寄存器 IDTR(Interrupt Descriptor Table Register)：48 位寄存器，与 GDTR 的作用类似，IDTR 寄存器用于存放中断描述符表 IDT 的 32 位线性基地址和 16 位表长度值。指令 LIDT 和 SIDT 分别用于加载和保存 IDTR 寄存器的内容。在机器刚加电或处理器复位后，基地址被默认地设置为 0，而长度值被设置成 0xFFFF。
- 局部描述符表寄存器 LDTR(Local Descriptor Table Register)：16 位寄存器，LDTR 寄存器中用于存放局部描述符表 LDT 的 32 位线性基地址、16 位段限长和描述符属性值。指令 LLDT 和 SLDT 分别用于加载和保存 LDTR 寄存器的段描述符部分。包含 LDT 表的段必须在 GDT 表中有一个段描述符项。当使用 LLDT 指令把含有 LDT 表段的选择符加载进 LDTR 时，LDT 段描述符的段基地址、段限长度以及描述符属性会被自动地加载到 LDTR 中。当进行任务切换时，处理器会把新任务 LDT 的段选择符和段描述符自动地加载进 LDTR 中。在机器加电或处理器复位后，段选择符和基地址被默认地设置为 0，而段长度被设置成 0xFFFF。
- 任务寄存器 TR(Task Register)：16 位寄存器，TR 寄存器用于存放当前任务 TSS 段的 16 位段选择符、32 位基地址、16 位段长度和描述符属性值。它引用 GDT 表中的一个 TSS 类型的描述符。指令 LTR 和 STR 分别用于加载和保存 TR 寄存器的段选择符部分。当使用 LTR 指令把选择符加载进任务寄存器时，TSS 描述符中的段基地址、段限长度以及描述符属性会被自动加载到任务寄存器中。当执行任务切换时，处理器会把新任务的 TSS 的段选择符和段描述符自动加载进任务寄存器 TR 中。

六、控制寄存器[9](Control Registers)

32 位的控制寄存器(CR_0~CR_3) 保存全局性、和任务无关的机器状态，用于控制和确定处理器的操作模式以及当前执行任务的特性，如下图所示：

---

[8] 陈光军，傅越千．微机原理与接口技术[M]．北京，北京大学出版社，2007：31.
[9] feng.qi．指令指针寄存器和标志寄存器[DB/OL]．https://www.cnblogs.com/feng-qi/articles/3055729.html，2013-05-03.

■ CR_0 中包含了 6 个预定义标志，0 位是保护允许位 PE(Protection Enable)，用于启动保护模式，如果 PE 位置 1，则保护模式启动，如果 PE=0，则在实模式下运行。1 位是监控协处理位 MP(Monitor Coprocessor)，它与第 3 位一起决定：当 TS=1 时操作码 WAIT 是否产生一个“协处理器不能使用”的出错信号。第 3 位是任务转换位(Task Switch)，当一个任务转换完成之后，自动将它置 1。随着 TS=1，就不能使用协处理器。CR_0 的第 2 位是模拟协处理器位 EM (Emulate Coprocessor)，如果 EM=1，则不能使用协处理器，如果 EM=0，则允许使用协处理器。第 4 位是微处理器的扩展类型位 ET(Processor Extension Type)，其内保存着处理器扩展类型的信息，如果 ET=0，则标识系统使用的是 287 协处理器，如果 ET=1，则表示系统使用的是 387 浮点协处理器。CR_0 的第 31 位是分页允许位(Paging Enable)，它表示芯片上的分页部件是否允许工作。由 PG 位和 PE 位定义的操作方式如下表所示：

PG	PE	方式
0	0	实模式，
0	1	未开启分页机制的保护模式
1	0	不存在
1	1	开启了分页机制的保护模式

■ CR_1 是未定义的控制寄存器，供将来的处理器使用。

■ CR_2 是页故障线性地址寄存器，保存最后一次出现页故障的全 32 位线性地址。

■ CR_3 是页目录基址寄存器，保存页目录表的物理地址，页目录表总是放在以 4K 字节为单位的存储器边界上，因此，它的地址的低 12 位总为 0，不起作用，即使写上内容，也不会被理会。CR_3 中含有页目录表物理内存基地址，因此该寄存器也被称为页目录基地址寄存器 PDBR(Page-Directory Base address Register)。

七、调试寄存器[10][11](Debug Registers)

32 位处理器有 8 个 32 位的调试寄存器 DR_0~DR_7，调试寄存器具有如下功能：

■ 设置发生断点的地址（线性地址）。

■ 设置断点的长度（可以为 1、2、4 个字节，但是执行断点只能是 1 个字节）。

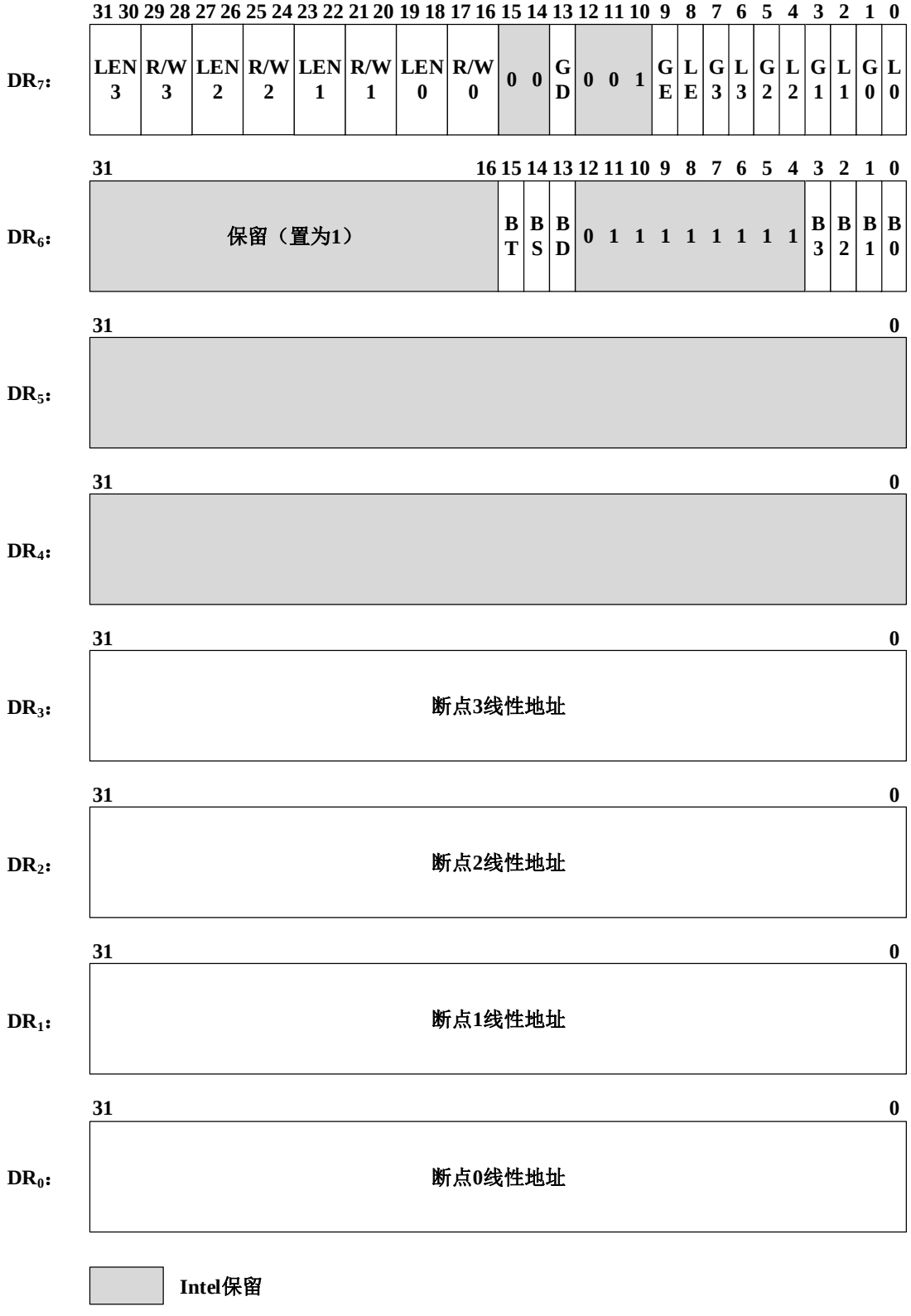
---

[10] Tweek．[总结]调试寄存器 原理与使用：DR0-DR7[DB/OL]．https://bbs.pediy.com/thread-107515.htm，2010-02-21．

[11] halfdead．Mistifying the debugger, ultimate stealthness[J/OL]．https://www.docin.com/p-36299796.html，2009-12-06．

- 设置在调试异常产生的地址执行的操作。
- 设置断点是否可用。
- 在调试异常产生时，调试条件是否是可用。

其中 DR_0~DR_3 这四个寄存器用于存储线性断点地址，DR_4 和 DR_5 是 Intel 保留寄存器，DR_6 主要是在调试异常产生后，报告产生调试异常的相关信息，DR_7 包含一些控制位，用于控制断点的方式。示意图如下：



下面详细介绍部分调试寄存器的具体作用：

1. 调试地址寄存器 DR_0~DR_3：

每个调试地址寄存器保存一个 32 位的断点线性地址。断点比较发生在物理地址转换之前。调试地址寄存器无论有没有开启分页机制，都是有效的，塔里木存储的是线性地址。如果开启了分页机制，线性地址会通过处理器的分页处理单元被转化为物理地址。如果没有开启分页机制，线性地址就等同于物理地址。

但是需要注意的是当分页开启的时候，不同的任务有不同的线性地址到物理地址的映射。在这种情况下，调试寄存器中的地址会被关联到其中一个任务上。由于这个原因，80386 在 DR_7 寄存器中有全局和局部开启位。这些位表明了调试地址是全局关联的（所有的任务）还是局部关联的（当前任务）。

2. 调试寄存器 DR_4~DR_5：

当调试扩展被开启的时候（控制寄存器 CR_4 中的 DE 标志置位），保留调试寄存器 DR_4 和 DR_5。任何试图引用这两个寄存器的行为都将会触发一个无效机器码异常。DE 不置位时，DR_4 和 DR_5 是 DR_6 和 DR_7 的别名。

3. 调试状态寄存器 DR_6：

这个寄存器主要是在调试异常产生后，报告产生调试异常的相关信息，处理器从不清空该寄存器的内容。该寄存器标志位显示下列信息：

- B0~B3 指明检测到断点条件。如果其中某个位被置位，则表示是相应的 DR_0~DR_3 断点引发了调试陷阱。此外还存在着一种情况，不管 DR_7 寄存器中的 Gi、Li(i=0,1,2,3) 如何设置，Bi 都会被置位。因此，只要是遇到 DR_i 指定的断点，总会设置 Bi，如果看到多个 Bi 置位，则可以通过 Gi、Li 的情况判断究竟是哪个调试寄存器引发的调试陷阱。
- BD 位（调试寄存器访问检测）指明指令流中的下一个指令将会访问某个调试寄存器，当调试控制寄存器 DR_7 中的 GD(General Detect)标志置位时才会生效。
- BS 位（单步）指明调试异常是由单步执行模式触发的。
- BT 位（任务切换）指明调试异常是任务切换导致的，目标任务的 TSS 中调试陷阱标志置位。

4. 调试控制寄存器 DR_7：

调试控制寄存器可以开启或关闭断点，并设置断点条件。它的标志和域控制以下信息：

- L0~L3 置位时（局部断点有效），启动断点模式，即激活当前任务的相关断点。当检测到断点条件，对应的 Li 位置位，产生调试异常。处理器在任务切换时自动清空这些标志，以避免一个不必要的断点出现在新任务中。
- G0~G3 置位时（全局断点有效），启动断点模式，即激活所有任务的相关断点。当检测到断点条件，对应的 Gi 位置位，产生调试异常。处理器在任务切换时不清空标志，这样断点可以出现在所有的任务中。
- LE 和 GE（局部和全局精确断点）开启后处理器能够精确定位引发数据中断条件的那条指令。
- GD 用于保护 DR_i，如果 GD 位为 1，则对 DR_i 的任何访问都会产生调试异常，并导致进入 1 号调试陷阱，即 IDT 的对应入口，这样可以保证调试器在必要的时候完全控制 DR_i。当检测到对 DR_i 寄存器的访问时，调试状态寄存器 DR_6 中的 BD 就会被置位，并产生一个异常。
- R/W0~R/W3 指明对应断点的断点条件，是读、写还是执行断点或是 I/O 端口断点。
- LEN0~LEN3 表示长度。

八、测试寄存器[12](Test Registers)

80386 设置了 8 个 32 位的测试寄存器 TR_0 到 TR_7，其中 TR_0~ TR_5 由 Intel 公司保留，用户只能访问 TR_6、TR_7，这两个寄存器用于在转换旁视缓冲器(Translation Lookaside Buffer)中测试随机存储器(RAM)和相联存储器(CAM)。TR_6 是测试控制寄存器，其内存放测试控制命令，TR_7 是测试状态寄存器，其内保存转换旁路缓冲器测试的数据。

[12] zmcomputer．调试寄存器和测试寄存器
[DB/OL]．https://blog.csdn.net/zmcomputer/article/details/5908227，2010-09-26．