

IBM360 系統工作原理

电子计算机参考資料編輯部

一九七二年十月

目 录

第 一 章 IBM/360 系统	1
1.1 通用性	1
1.2 兼容性	2
1.3 程序逻辑	2
1.4 控制与检验手段	2
1.5 多系统工作	3
1.6 输入输出	3
1.7 工艺	3
第 二 章 系统结构	4
2.1 主存贮器	4
2.1.1 信息格式	4
2.1.2 编址	5
2.1.3 信息的安排	6
2.2 中央处理机	6
2.2.1 公共寄存器	7
2.2.2 浮点寄存器	7
2.3 算术和逻辑部件	7
2.3.1 定点操作	8
2.3.2 十进制运算	8
2.3.3 浮点操作	9
2.3.4 逻辑操作	10
2.4 程序的执行	12
2.4.1 指令的信息格式	12
2.4.2 地址形成	14
2.4.3 顺序排列的指令的执行	15
2.4.4 转移	15
2.4.5 程序状态字	16
2.4.6 中断	17
2.5 存贮保护手段	20
2.6 计时器	20
2.7 直接控制手段	20
2.8 多系统工作手段	21
2.9 输入输出	21
2.9.1 输入输出设备和控制器	21
2.9.2 输入输出设备的共轭	21

2.9.3	通道	22
2.9.4	GPU 输入输出指令	22
2.9.5	输入输出操作的开始	23
2.9.6	通道指令	24
2.9.7	输入输出操作的完成	25
2.9.8	输入输出中断	25
2.10	控制台	26
2.10.1	控制台的用途	26
2.10.2	操作员控制台	26
2.10.3	人工控制台	27
2.10.4	工程控制台	27
第 三 章	定点操作	28
3.1	数据的信息格式	28
3.2	数的表示法	29
3.3	结果特征	29
3.4	指令的信息格式	30
3.5	指令	31
3.5.1	取数	32
3.5.2	取半数	32
3.5.3	取数并检验	32
3.5.4	取补	33
3.5.5	取正	33
3.5.6	取负	34
3.5.7	成组取数	34
3.5.8	加法	35
3.5.9	半字加	35
3.5.10	代码加	36
3.5.11	减法	37
3.5.12	半字减	37
3.5.13	代码减	38
3.5.14	比较	39
3.5.15	半字比较	39
3.5.16	乘法	40
3.5.17	半字乘	40
3.5.18	除法	41
3.5.19	转换成二进制	41
3.5.20	转换成十进制	42
3.5.21	存	42
3.5.22	存半字	43

3.5.23	成组存	43
3.5.24	左移	44
3.5.25	右移	44
3.5.26	双字左移	45
3.5.27	双字右移	45
3.6	执行定点操作时的特殊情况	46
第四章	十进制算术	47
4.1	数据的信息格式	47
4.2	数的表示法	48
4.3	结果特征	48
4.4	指令的信息格式	49
4.5	指令	49
4.5.1	十进制加法	50
4.5.2	十进制减法	50
4.5.3	清除加	51
4.5.4	十进制比较	52
4.5.5	十进制乘法	52
4.5.6	十进制除法	53
4.5.7	装配	54
4.5.8	拆卸	54
4.5.9	移位传送	55
4.6	执行十进制算术指令时的特殊情况	55
第五章	浮点操作	57
5.1	数据的信息格式	57
5.2	数的表示法	58
5.3	规格化	58
5.4	结果特征	59
5.5	指令的信息格式	59
5.6	指令	60
5.6.1	取数	61
5.6.2	取数并检验	62
5.6.3	取补	62
5.6.4	取正	63
5.6.5	取负	63
5.6.6	规格化加法	64
5.6.7	不规格化加法	65
5.6.8	规格化减法	66
5.6.9	不规格化减法	67
5.6.10	比较	68

5.6.11 对分	69
5.6.12 乘法	69
5.6.13 除法	71
5.6.14 存	72
5.7 执行浮点操作时的特殊情况	72
第 六 章 逻辑操作	74
6.1 数据的信息格式	74
6.2 结果特征	75
6.3 指令的信息格式	76
6.4 指令	77
6.4.1 传送	78
6.4.2 数字传送	79
6.4.3 区传送	79
6.4.4 代码比较	79
6.4.5 与	80
6.4.6 或	81
6.4.7 非或	82
6.4.8 屏蔽检验	83
6.4.9 读字符	83
6.4.10 存字符	84
6.4.11 送地址	84
6.4.12 重编码	85
6.4.13 重编码与检验	85
6.4.14 编辑	86
6.4.15 编辑与说明	88
6.4.16 逻辑左移	89
6.4.17 逻辑右移	90
6.4.18 双字逻辑左移	90
6.4.19 双字逻辑右移	90
6.5 逻辑操作时的特殊情况	91
第 七 章 转移	92
7.1 正常操作序列	92
7.1.1 引起破坏正常指令序列的特殊情况	93
7.2 方向的选择	94
7.3 指令的信息格式	94
7.4 转移指令	95
7.4.1 条件转移	96
7.4.2 返回转移	96
7.4.3 按计数器转移	97

7.4.4	变址数大于时转移	97
7.4.5	变址数小于或等于时转移	98
7.4.6	执行	98
7.5	转移时的特殊情况	99
第 八 章	状态转换	102
8.1	程序状态	102
8.1.1	课题状态	102
8.1.2	等待状态	103
8.1.3	中断屏蔽状态	103
8.1.4	停机状态	104
8.2	存贮保护	105
8.2.1	赋键	105
8.2.2	存贮保护破坏时执行的动作	105
8.2.3	被保护单元	105
8.3	程序状态字	106
8.4	多系统工作	107
8.4.1	直接编址的存贮器单元的移动	107
8.4.2	机器出错信号	108
8.4.3	送入初始程序的外部激励	108
8.5	指令的信息格式	108
8.6	指令	109
8.6.1	送 PSW	109
8.6.2	建立程序屏蔽	110
8.6.3	建立系统屏蔽	110
8.6.4	访问管理程序	111
8.6.5	建立存贮键	111
8.6.6	读存贮键	111
8.6.7	直接写入	112
8.6.8	直接读出	112
8.6.9	诊断	113
8.7	状态转换时的特殊情况	114
第 九 章	中断	115
9.1	中断时的动作	115
9.1.1	指令的执行	116
9.1.2	中断源的识别	117
9.1.3	指令位置的确定	117
9.2	输入输出中断	118
9.3	程序中断	118
9.3.1	操作码错误	119

9.3.2	优先权操作	119
9.3.3	执行指令错误	119
9.3.4	存贮保护破坏	119
9.3.5	不正确编址	119
9.3.6	不正确说明	120
9.3.7	不正确数据	120
9.3.8	定点溢出	120
9.3.9	定点除法错误	120
9.3.10	十进制溢出	121
9.3.11	十进制除法错误	121
9.3.12	阶溢出	121
9.3.13	阶消失	121
9.3.14	意义丧失	121
9.3.15	浮点除法错误	121
9.4	访问管理程序时的中断	121
9.5	外中断	122
9.5.1	计时器	122
9.5.2	中断按钮	123
9.5.3	外部信号	123
9.6	机器检验电路的中断	124
9.7	中断的优先权	124
9.8	中断时的特殊情况	125
第 十 章	输入输出操作	126
10.1	输入输出设备的连接	126
10.1.1	输入输出设备	126
10.1.2	控制器	126
10.1.3	通道	127
10.1.4	系统工作	129
10.1.5	操作的兼容性	130
10.2	输入输出设备的控制	131
10.2.1	输入输出设备的编址	131
10.2.2	执行输入输出指令时的特殊情况	133
10.2.3	输入输出系统的状态	133
10.2.4	输入输出系统清除成初始状态	135
10.2.5	结果特征	136
10.2.6	指令的信息格式	138
10.2.7	输入输出指令	139
10.2.8	开始输入输出	139
10.2.9	查询输入输出	141

10.2.10	停止输入输出	142
10.2.11	查询通道	144
10.3	输入输出操作的执行	144
10.3.1	数据组	145
10.3.2	通道地址字	145
10.3.3	通道控制字	145
10.3.4	通道指令码	147
10.3.5	存贮区的确定	148
10.3.6	链	149
10.3.7	封锁写入	151
10.3.8	程序控制中断	151
10.3.9	指令	152
10.4	输入输出操作的结束	156
10.4.1	操作结束的方法	156
10.4.2	输入输出中断	160
10.4.3	通道状态字	162
10.4.4	设备状态的信号	163
10.4.5	通道状态的信号	167
10.4.6	通道状态字的内容	170
10.4.7	计数器	172
10.4.8	状态位	174
第十一章	系统控制台	175
11.1	系统控制台的用途	175
11.1.1	系统清除	175
11.1.2	信息的写人与显示	176
11.1.3	原始程序的输入	176
11.2	操作员控制台	177
11.2.1	紧急开关	178
11.2.2	加电按钮	178
11.2.3	断电按钮	178
11.2.4	中断按钮	178
11.2.5	等待灯	178
11.2.6	人工控制灯	178
11.2.7	系统灯	179
11.2.8	检验灯	179
11.2.9	输入灯	179
11.2.10	输入开关	179
11.2.11	输入按钮	180
11.2.12	前缀选择开关	180

11.3	人工控制台	180
11.3.1	系统清除按钮	180
11.3.2	停机按钮	181
11.3.3	工作各种开关	181
11.3.4	起动按钮	181
11.3.5	存储器类型开关	181
11.3.6	地址开关	182
11.3.7	数据开关	182
11.3.8	写入按钮	182
11.3.9	显示按钮	182
11.3.10	置指令地址按钮	182
11.3.11	比较开关	182
11.3.12	附加前缀灯	183
11.4	工程控制台	183
附 录		184
附录一	指令应用例题	184
附录二	定点操作与数的补码表示法	206
附录三	浮点操作	207
附录四	2 为底的指数表(略)	209
附录五	十六进制数和十进制数转换表(略)	209
附录六	EBCDIC 与 ASCII-8 代码表	210
附录七	信息格式与表格	211
附录八	系统结构的补充	253

第一章 IBM/360 系统

IBM/360 系统是供处理数据用的半导体计算机系列，它具有满足现代经济、科学和技术上要求的那种速度、精确度和工作灵活性。系统各种型号机器之间在程序上是兼容的。系统中采用了微电子线路的最新工艺成就，从而把系统的运算效率提到更高水平，也提高了系统的灵活性和可靠性。这又给通过新途径解决各个领域的信息处理问题提供了可能性，使得实现起来更经济，工作起来更方便。IBM/360 系统是处理数据用的既统一而又相互联系的一种新系统，其目的是以从目前和将来应用的角度来看更有前途的结构来代替陈旧的逻辑结构。

IBM/360 的逻辑结构可在保存程序‘向上’和‘向下’兼容的条件下有效地用设备实现不同水平的运算率。对于超高速的运算率和特高的可靠性的要求，可以通过用专门的多道系统手段把几种型号机器联为一个多系统来得到满足。

1.1 通用性

IBM/360 系统是为了用于经济与科学的目的以及解决信息传送和控制问题而设计的通用系统。系统的基本计算能力由标准指令系统提供。此指令系统可以附加上处理十进制数据的手段，以形成经济用指令系统；当增加了浮点操作时，则可得到科学用指令系统。若是给经济用与科学用指令系统添上存贮保护设备的话，就形成通用指令系统。为了远距离处理数据以及输入分配操作和实时操作，系统可以接上直接控制设备和电子钟。

IBM/360 系统能够装备大容量的编址存贮器。如果併上中型快速存贮器和大型中速存贮器，360 系统存贮器的容量比现有的可以显著地扩大。所以借助系统里的存贮器部件的层次结构既能满足运算效率的要求，又可满足大存贮容量的要求。如果以后有必要，系统存贮容量尚可进一步扩充。

IBM/360 系统采用了标准的方式，以连接在用途、数据传送速度和访问时间方面各不相同的各种输入输出设备。

IBM/360 系统的各种型号机器是在系统的各组成部件经过以下的选择之后得到的，这就是要使对内部运算率、函数功能以及输入输出设备提出的为数众多而又相互矛盾的要求得到最充分的满足。

IBM/360 系统的不同型号机器在以下几方面各不相同：(1)存贮器的操作速度；(2)存贮器的物理位数*（即每次访问时所得数据量）；(3)寄存器的位数；(4)并行处理信息的能力。尽管如此程序员从逻辑上感觉不到这些差异。由于有好几台中央处理机(CPU)，故内部运算能力可以广泛地加以选择。其范围：能力最强的型号与能力最差的型号内部运算能力的比例，就科学计算来说，前者大约是后者的 50 倍，就经济计算来说，大约是 15 倍。

*：本文将 Физический 译为“物理学的”等。有些地方可理解成“实际的”。

1.2 兼容性

IBM/360 系统的各种型号机器都是‘向上’和‘向下’兼容的，也就是说任何一道程序在任何一种型号机器上都得出同样的结果。兼容性便于系统扩充、维护方便而且掌握简单。

兼容原则有三条限制：

1. 对程序而言，系统的能力在任何场合都应一样。例如各 GPU 都要具有相同的一整套的必须的（即根据用户选择来规定的）能力；同样地，在存贮容量、输入输出设备的数量、类型和优先权等方面也应等价。

2. 程序不应依赖于 GPU 中的指令执行时间、数据输入或输出速度以及输入输出通道执行指令的时间。

3. 兼容原则不适用于那些细微的操作，它们出现的频率和利用结果的特点在不同型号机器中是不一样的。这些细微操作涉及到机器执行各种不许可的程序，即有差错的程序，以及机器的故障，我们以后将详细介绍。

1.3 程序逻辑

在 IBM/360 系统中，设备与程序的相互作用问题具有重要意义。系统设计成能供管理程序操作作用，也就是供这样一种程序操作作用的：它能调配与执行一切输入输出指令，处理特殊情况，实现协调以及保证几道程序的执行。IBM/360 系统能从一道程序有效地转换到另一道程序，及使存贮器中的程序搬场。在一般程序员看来，管理程序和设备是一个统一的整体。

1.4 控制与检验手段

当系统外部输入输出设备及 GPU 本身产生某种条件时，中断系统能自动改变 GPU 的状态。中断使 GPU 从一道程序转换到另一道程序，改变的不仅是指令地址而且是有关机器状态的所有主要信息。

存贮保护手段赋与一道程序，在某种情况下，即当另一道程序错误地试图在分给前者的存区写入信息时，以保护自己的存区。存贮保护的存在不会引起工作效率的任何损失。存贮保护既适用于 GPU 操作，也适用于由输入输出通道实现的操作。

在执行程序时要对指令和数据的正确性进行检验。这种检验能发现程序出错和机器的故障并把它们彼此区分开来。所以程序出错不会引起机器故障信号的出现，因为每种出错类型都引起不同的中断。在机器发生故障时，除了引起中断外，在预先给定的存贮单元自动写入供识别故障所必须的信息。这就大大缩短了排除机器故障所必须的平均时间。此外，依靠把手动操作减到最少限度以缩小操作员所带来的差错。为了减少操作人员的意外差错。控制台被看成是输入输出设备，这些控制台的工作是在管理程序控制之下实现的。

1.5 多系统工作

IBM-360 系统几种型号机器可以连成一个整体。在各 CPU 之间可以实现三种水平的联结。在利用公用输入输出设备,例如磁盘存贮器时可以获得容量最大和操作较快的联结。当两个独立的系统通道之间直接相连时,信息传送更快。最后,在某些型号机器里两台 CPU 可有一台公用内存贮器,从而能以存贮器操作的速度来交换信息。除了这些联结方式之外,还可以让一台 CPU 中断另一台的工作并获得后者的状态信息。

1.6 输入输出

CPU 和输入输出设备之间的信息沿输入输出通道传送,CPU 和输入输出设备均设有相应的控制手段。通常,通道与 CPU 是异步工作的。每条通道可由数条子通道组成。此时,一条通道为若干台慢速操作的输入输出设备,例如穿孔卡片读出机、穿孔机、打印机和终端装置等服务。这种通道叫做多通道。只有一条子通道的通道比起多通道可能以更高的速度进行工作,它叫选择通道。在上述两种场合,信息均按字节(即按 8 位)从输入输出设备送入通道。为了控制所有的通道和子通道,要利用相同的 CPU 指令和通道指令。

每台输入输出设备通过“共轭”与一条或几条通道连线。“共轭”既可接现有的输入输出设备又可接以后研制出来的输入输出设备,而且不改变指令系统或通道工作的特性。当必须协调输入输出设备与“共轭”的联系时可利用控制器。一台控制器或输入输出设备可接两条通道,这就为 CPU 沿任何一条通道与输入输出设备连线提供了可能性。

1.7 工艺

IBM/360 系统采用固体集成电路,以保证系统工作的高度可靠。此外,这种电路比已有的电路工作速度快、体积小,且可自动化生产。

第二章 系统结构

IBM/360 系统的基本结构由主存贮器、中央处理机、选择通道与多通道以及通过控制器接至通道的输入输出设备所构成。各系统具有通过公共输入输出设备、通道或公共存贮器彼此互连的能力。图 1 示出一个系统结构的基本方案。

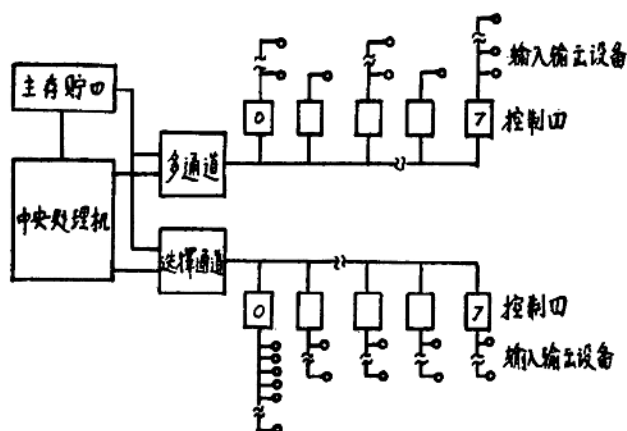


图 1 IBM/360 系统的基本逻辑结构

2.1 主 存 贮 器

在结构上，存贮部件或跟 GPU 成为一个整体，或成为单独的装置。存贮周期与 GPU 的工作周期没有严格的联系，从而能以最佳方式选择在给定字长条件下的存贮器工作速度。不同型号机器的主存贮器物理实现方法的差别并未在系统的逻辑结构上有所反映。

GPU 访问存贮器与输入输出设备的工作无关。如果在输入输出操作时要访问 GPU 执行操作用的存贮单元，那么，访问是按照它们到达的顺序来实现。如果第一个访问改变了单元内容，那末后面对这个单元的访问均获得新的内容。当输入输出设备与 GPU 同时访问同一个单元时绝不会引起机器错误信号的出现。

2.1.1 信 息 格 式

信息在系统的主存贮器与 GPU 部件之间按 8 位或 8 位的倍数进行传送。八位信息称为字节，它是所有信息格式结构的基本单位。随字节传送的第九位叫偶么位或叫检验位。偶么位的值不能用程序方法来改变；它的唯一用处是在发现偶么位出错时引起程序中断。所以在描述信息场和寄存器的长度时偶么检验位不在考虑之列。存贮器的容量由字节的数量来表示，

并不决定于存储器的物理位数多少。

字节可以一个个地单个处理,或是按几个字节组成的场来处理。半字是由两个字节顺序排列而成的一个组,它是指令的基本组成单位。字是四个字节顺序排列而成的一个组;双字则是由两个字所构成的一个场。(见图2)。任何一个场或一字节组的位置都由最左边一个字节的地址给定。

场的长度或由应执行的操作码确定(隐含给定字长)或者借助于指令中专门的位来指明(显式给定字长)。场的长度为隐含给定时,则是说信息有固定的长度,它可相当于1、2、4或8个字符。

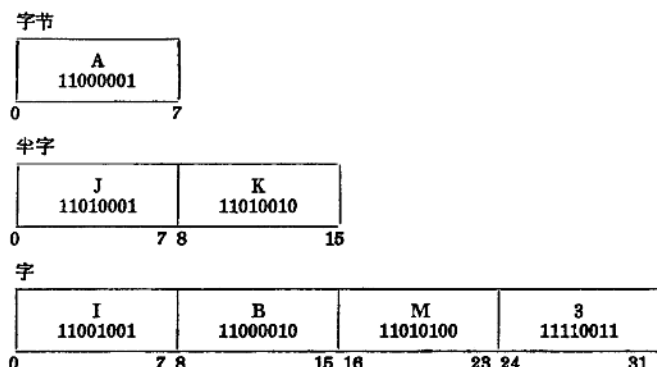


图2 信息格式示例

场的长度是显式给定时,则是说信息具有可变长度。各个可变长的操作数在长度上相差若干个字节。

在指令的任何信息格式之内或在任何固定长操作数中,信息格式的各位从0开始从左向右顺序编号。

2.1.2 编 址

包含字节的存储器单元从0开始顺序编号;每一编号视为相应字节的地址。在字节组编址时,存储器指明这组最左字节的地址。在24位字长的地址码内,可编址的字节数最多可为16777216。这组主存储器地址包括了某些专用单元地址。

存储器中的编址是循环进行的。跟在最大地址16777215存储单元之后的是地址为0的单元。可变长操作数可以一部分位于存储器的最后一些单元里,而另一部分位于前面一些单元里,此时也是按照一般方法加以处理。

如果机器装上小于最大容量的存储器,则编址从0开始顺序进行。关于把操作数全部或部分置于该设备存储范围之外的试图被看作是不正确的编址。只当数据确实被利用时才会有不正确编址。关于在数据被利用之前就要完成的操作不划为不正确编址。

有些型号机器的主存储器可以被几台CPU共同使用。这时,任何一个单元对于不同的CPU都有一个相同的普通地址。

2.1.3 信息的安排

固定长度的场，例如半字或双字应从该信息组**整数边界**开始安排在主存储器中。当某信息组的存储地址是信息组字节数的整数倍时，它的边界便叫做**整数边界**。例如，一般字（4个字节）在存储器里应该安排得使它们的地址是4的倍数。半字（2个字节）地址应是2的倍数，而双字（8个字节）则要有相当于8的倍数的地址。

存储器地址以二进制代码表示。在使用二进制代码的情况下，对应半字、一般字和双字整数边界的二进制地址分别有一个、两个或三个低位等于0（见图3）。例如，一般字的整数边界就是两个低位等于0的二进制地址。

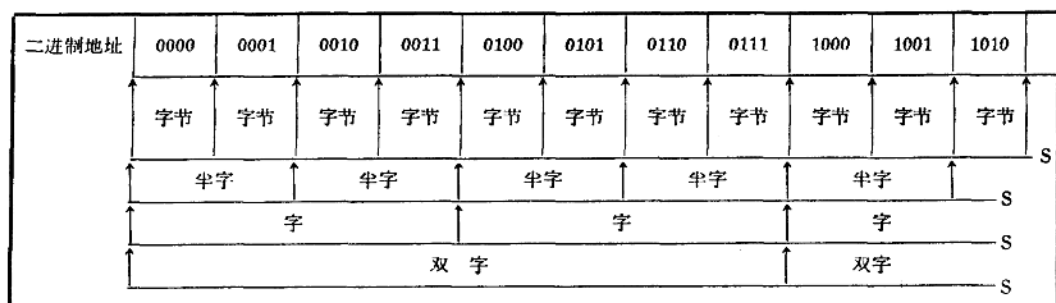


图3 半字、字和双字的边界

可变长度的场没有整数边界，可从任何字符开始。

2.2 中央处理机

中央处理机(图4)具有供以下用途的设备：主存储器的编址、信息的选取和写入、数据的算术处理和逻辑处理、保证所需的指令次序，还有使存储器 and 外部设备进行信息交换的设备。

执行指令时由系统的控制器对CPU进行控制。虽然IBM/360系统不同型号机器中控制器物理实现的方法可能不同，但各种型号机器中控制器的逻辑功能都一样。

CPU有16个供定点操作数用的公共寄存器和4个供浮点操作数用的寄存器。物理上这些寄存器可用有源元件制成单个存储部件形式，或者是利用一部分主存储器。在任何情况下，公共寄存器的地址和功能均相同。

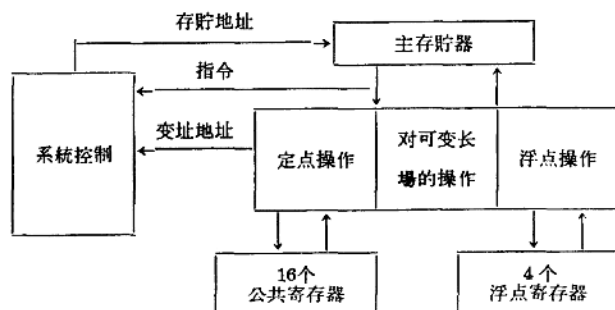


图4 中央处理机

2.2.1 公共寄存器

CPU 同 16 个公共寄存器打交道, 在地址运算和变址操作中, 公共寄存器可作为变址寄存器。还可在逻辑操作和定点算术操作中用来存放数据。寄存器可放一个字(32 位)。公共寄存器编号自 0 到 15, 并借助指令中的 4 位的场进行选取。在指令里此场用字母 R 表示(图 5)。

场 R	寄存器编号	公共寄存器	浮点寄存器
0000	0	32位	64位
0001	1		
0010	2		
0011	3		
0100	4		
0101	5		
0110	6		
0111	7		
1000	8		
1001	9		
1010	10		
1011	11		
1100	12		
1101	13		
1110	14		
1111	15		

图 5 公共寄存器与浮点寄存器

在执行某些操作时, 相邻的两个寄存器可以联合使用, 以进行双倍字长操作。这时, 编址寄存器存放操作数的高位部分且地址应是偶数, 而另一个存放操作数低位部分的作为附加寄存器, 它要有奇数地址。

2.2.2 浮点寄存器

规定 4 个寄存器作为浮点操作。这种寄存器有两个字长(64 位)。浮点寄存器里可以放置短操作数(一个字长)或长操作数(两个字长)。短操作数放置于寄存器的高位, 低位则忽略不计, 且它们的状态在执行一般精度的运算时不起变化。浮点操作寄存器的编号为 0、2、4、6(图 5)。操作中使用的寄存器类型由指令操作码决定。

2.3 算术和逻辑部件

算术和逻辑部件可以处理固定长的二进制数(包括整数和浮点数)、可变长的十进制整数以及固定长或可变长的逻辑信息。工作方式既可以并行, 也可串行。在算术部件的规模, 移位链数量, 完成不同运算的可互换程度方面, 各 CPU 有所不同, 但这并不影响对别的方面和

操作的逻辑方面。

GPU执行的算术操作和逻辑操作分为四类：(一)定点操作；(二)十进数操作；(三)浮点操作；(四)逻辑操作。这几种操作的差别是它们所用数据信息格式不同、因而操作中所用寄存器也就不同、还区别于操作的指令组以及给定场的长度的方法也不同。

2.3.1 定 点 操 作

基本的算术操作数是 32 位的定点二进制字。为了提高运算率或改进存贮器的负载，对多数操作，都可给定半字长的 16 位操作数(图 6)。但为了保持精确度起见，有些乘积与所有被除数均为 64 位长。

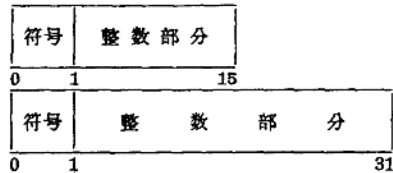


图 6 定点数的信息格式

由于 32 位的字长内可以放 24 位的地址码，故定点操作不但能够用于对整数的运算而且能够用于对地址的运算。这样的综合利用可以节约设备，并且在进行地址运算时可以使用全部定点操作指令和某些逻辑操作。例如，对地址诸成分可以进行乘法操作、移位操作和任何逻辑转换。

地址成分和操作数最好用补码表示。因为这样操作数的长度便于伸缩。由于在处理整数和形成地址的算法中往往必须重复地访问操作数和中间结果，所以使用几个寄存器在串行算术操作和地址计算时有着重大优点。

对于一个在寄存器而另一个在内存或也在寄存器的操作数，可进行加、减、乘、除和比较操作。由于数以补码表示而且能够从一个字向另一个字固定进位，从而简化了提高精度的操作。放在一个寄存器里的字或放在两个相邻寄存器里的双倍长的字均可左移或右移。两条转换指令：“转换成二进制”(CONVERT TO BINARY)和“转换成十进制”(CONVERT TO DECIMAL)能够实现十进制与二进制之间的转换，而不需查表。寄存器成组取数指令和寄存器成组写入指令能够卓有成效地实现子程序的转换。

2.3.2 十 进 制 算 法

对于那些在原始数据输入和结果输出之间计算量不大的加工规定有十进制数的操作。这类加工形式在经济计算，特别是在使用面向问题的语言时常常碰到。鉴于对单个信息单位执行的算术操作数量有限，从十进制转换成二进制然后再转换成十进制未予考虑，而且由于经常访问存贮器故利用寄存器存放中间结果也没什么优越性，所以增加了一种两个操作数和结果都位于存贮器的对十进制数的操作，也就是“存贮器——存贮器”型的操作。对十进制数的操作包括有加、减、乘、除和比较操作。

十进制数作为有符号整数来处理，格式是可变长度的。十进制操作数的最大长度为 16