1. **HW1-组合逻辑：**
2. 化简逻辑表达式：。

解:

1. 化简逻辑表达式

解:

1. 化简逻辑表达式：

解：

1. 化简逻辑表达式：

解：

1. 化简逻辑表达式：

解:

根据卡诺图，可得：

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ABC  DE | 000 | 001 | 011 | 010 | 110 | 111 | 101 | 100 |
| 00 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 01 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 11 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 10 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |

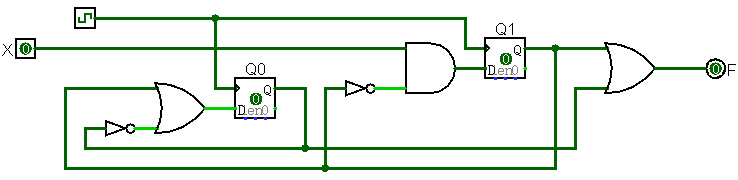
**2.HW2-时序逻辑:**

1. 某序列检测器，输入为data，输出为Q。当检测到1010时，Q输出1个周期的1，否则Q输出0。给出状态机次态逻辑表达式和Q的逻辑表达式。注意：{1010}为独立检测，不与后续序列拼接，即{101010}视为1次匹配成功。

答案：假设寄存器D2D1D0的编码000、001、010、011及100分别对应S0至S4。由于Q输出宽度以周期为单位，因此Q只能是状态的函数，即该状态机是摩尔型状态机。



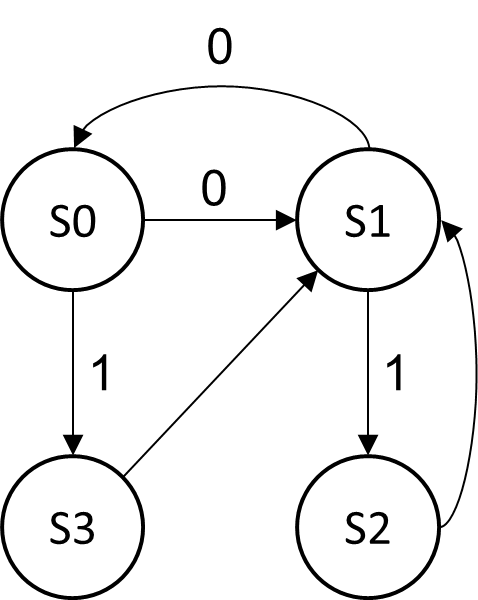
1. 状态机如下图所示。构造次态逻辑表达式和输出表达式，写出状态转换及输出表，画出状态图，分析状态机类型。



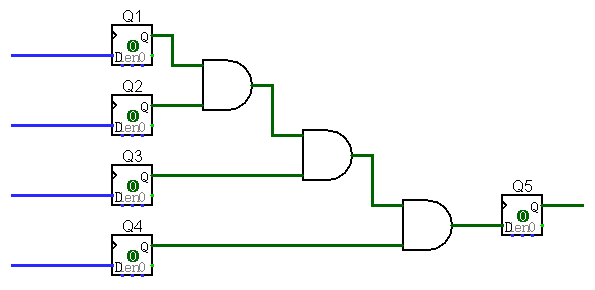
答案：

Moore型状态机，因为F只与状态寄存器相关。

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Q1 | Q0 | X | Q1n | Q0n | F |  | 现态 | X | 次态 |
| 0 | 0 | 0 | 0 | 1 | 0 |  | S0 | 0 | S1 |
| 0 | 0 | 1 | 1 | 1 | 0 |  | S0 | 1 | S3 |
| 0 | 1 | 0 | 0 | 0 | 1 |  | S1 | 0 | S0 |
| 0 | 1 | 1 | 1 | 0 | 1 |  | S1 | 1 | S2 |
| 1 | 0 | 0 | 0 | 1 | 1 |  | S2 | 0 | S1 |
| 1 | 0 | 1 | 0 | 1 | 1 |  | S2 | 1 | S1 |
| 1 | 1 | 0 | 0 | 1 | 1 |  | S3 | 0 | S1 |
| 1 | 1 | 1 | 0 | 1 | 1 |  | S3 | 1 | S1 |

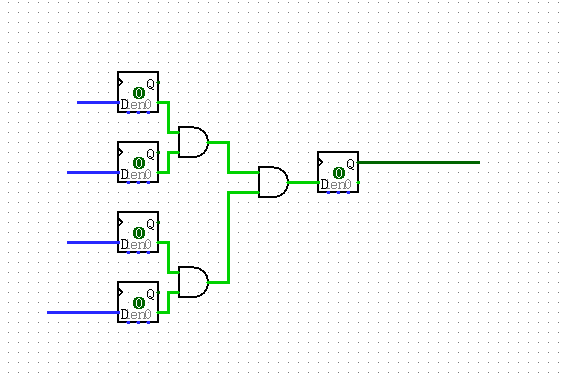


1. 对于如下电路，假设每个与门的延迟均为T，连线延迟、寄存器固有延迟等均为0。
   1. 计算关键路径的延迟以及寄存器的最大时钟频率。
   2. 重构电路以提高时钟频率，并计算提高比例。



延迟为3T。最大时钟频率为1/3T。

优化后延迟为2T。最大时钟频率为1/2T，比原设计频率提高50%。



1. 《数字设计和计算机体系结构》：第3.20题。

解：

这是一个Mealy型状态机，共有两位寄存器Q1和Q0，编码分别为：S0 = 2b00,

S1 = 2b01,S2 = 2b10

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 现态  S1S0 | A | B | 次态  S1S0 | F |
| 00 | 0 | X | 00 | 0 |
| 00 | 1 | X | 01 | 0 |
| 01 | X | 0 | 00 | 0 |
| 01 | X | 1 | 10 | 0 |
| 10 | 1 | 1 | 10 | 1 |
| 10 | 0 | 0 | 00 | 0 |
| 10 | 0 | 1 | 00 | 0 |
| 10 | 1 | 0 | 00 | 0 |

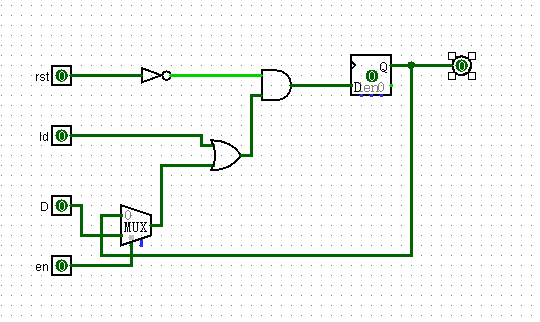
表达式如下：

1. 功能：检测A和B先后输入为1，此后AB同时为1。
2. 用D触发器为核心，设计一个支持使能(En若为0，则寄存器值不变)、预置(Ld为1，则寄存器值为1)、清除(Rst为1，则寄存器被清除)功能的寄存器。寄存器输入信号为D，输出为Q。所有控制信号都是同步控制信号。3个信号的优先级从高到低为：Rst、Ld、En。

解：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Rst | Ld | En | D | Q | D触发器输入 |
| 1 | X | X | X | X | 0 |
| 0 | 1 | X | X | X | 1 |
| 0 | 0 | 0 | X | Q | Q |
| 0 | 0 | 1 | 0 | X | 0 |
| 0 | 0 | 1 | 1 | X | 1 |

电路结构如下：



**3.HW3-数据表示与运算方法:**

1. 计算2223，2224，2225对应的十进制值。

解：

1. 分别计算0b1011101与0xB23的十进制值（均按无符号数处理）

解:

0b1011101 = +

1. 用16进制方式表示无符号数10011111102。

解：

1. 除了十进制转换为二进制的方法，下面给出了从十进制转换为五进制和九进制的2个具体案例。根据该案例总结出从10进制转换为N进制的一般性方法。

表2‑6 十进制2007转换为5进制的计算过程

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 步骤 | 被除数 | 商 | 余数 | 位序 | 备注 |
| 1 | 2007 | 401 | 2 | 0 | 2007除以5 |
| 2 | 401 | 80 | 1 | 1 | 401除以5 |
| 3 | 80 | 16 | 0 | 2 | 80除以5 |
| 4 | 16 | 3 | 1 | 3 | 16除以5 |
| 5 | 3 | 0 | 3 | 4 | 被除数小于除数5，计算结束 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | 4 | 3 | 2 | 1 | 0 | 位序 |
| 200710 | = | 3 | 1 | 0 | 1 | 2 | 5 |

表2‑7 十进制2018转换为9进制的计算过程

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 步骤 | 被除数 | 商 | 余数 | 位序 | 备注 |
| 1 | 2018 | 224 | 2 | 0 | 2018除以9 |
| 2 | 224 | 24 | 8 | 1 | 224除以9 |
| 3 | 24 | 2 | 6 | 2 | 24除以9 |
| 4 | 2 | 0 | 2 | 3 | 被除数小于除数9，计算结束 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | 3 | 2 | 1 | 0 | 位序 |
| 201810 | = | 2 | 6 | 8 | 2 | 9 |

解:10进制数转化为N进制数为将该数不断模N，将余数倒转串联起来就是转化后的结果

1. 给出快速估算228的方法。

解：

1. 给出6位二进制补码的表示范围。

**解：表示范围为: -32 ~ 31。**

1. 将下列十进制数转换为6位二进制补码并完成计算，同时指出结果是否存在溢出。

①16+15 ② 16+18 ③16-8 ④-16-16 ⑤-24-13

解：

**溢出**

**溢出**

1. 下列代码执行结束后，请用32位二进制补码方式分别表示c、s与us的值。

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | char c ;  short s ;  unsigned short us ;  c = -1 ;  s = c ;  us = (unsigned short)c ; |

解:

**c = 0xffff\_ffff**

**s = 0xffff\_ffff**

**us = 0x0000\_ffff**

4.HW4-指令与汇编:

1. 请从指令编码格式的角度分析MIPS寄存器个数为32个的合理性。提示：主要分析如果寄存器个数多于32时，对于指令编码会带来哪些负面影响。

**解：如果寄存器的个数多于32个，则编码位数超过5位，一条指令长度为32位，很多指令都不能够一次生效，如beq，跳转立即数的位数至少少2位，即最多14位，跳转范围小了很多，还有其他对于R型指令来说，填满3个寄存器需要15位以上的位数，这不利于指令的识别和增加。**

1. 请以指令为单位分析jal的被调用函数的入口地址的范围。

**解：jal跳转立即数总共有26位，跳转范围为2^26条指令，大约为4GB**

1. 请结合C语言的特点，分析beq指令格式中imm域的合理性。

**解：在C语言中，一条判断语句通常不会太长，而beq中imm共有16位，64K对于一条判断语句的语句体来说是足够使用了的。**

1. 请把下列C代码翻译为汇编代码。假设$s0存储着变量i，可用的寄存器只有$s0~$s3。

do {

循环体；

while ( 0<i && i<100 ) ;

解：

**Loop：**

**循环体**

**Ble $s0,$0,loop\_end**

**Bge $s0,100,loop\_end**

**J loop**

**Loop\_end:**

1. 请用beq和bne以及与分支无关的指令来完成下面这条语句的功能。提示：要防止溢出。

slt $s0, $s1, $s2

解：

**sub $t0, $s1, $s2**

**beq $t0, $0, SET0 // s1等于s2**

**srl $t0, $t0, 31 // 只保留符号位**

**beq $t0, $0, SET0 // 符号位为0：S1>S2**

**addi $s0, $0, 1 // 符号位为1：S1<S2**

**j END**

**SET0:**

**addi $s0, $0, 0**

**END:**

1. 程序员编写了如下汇编程序来完成1000个字节的复制任务。假设循环开始前，$s0和$s1分别指向源字符串首地址和目的字符串首地址。

|  |  |
| --- | --- |
| LOOP: | lb $t0, 0($s0)  beq $t0, $0, TAIL  sb $t0, 0($s1)  addi $s0, $s0, 1  addi $s1, $s1, 1  j LOOP |
| TAIL: |  |

1. 请计算上述代码执行的指令总数。
2. 请最大化优化上述代码。优化代码只能使用本章讲授的指令。优化后代码仍然采用循环结构，且每次循环只能有1次复制操作。
3. 请计算优化后代码执行的指令总数。

解：

**1：复制1个字节执行6条指令，1000个字节总共执行 6 \* 1000 + 2 = 6002条指令**

**2：**

**Addi $t1,$0,250**

**Loop:**

**Lw $t0,($s0)**

**Sw $t0,($s1)**

**Addi $s0,$s0,4**

**Addi $s1,$s1,4**

**Bne $t1,$0,loop**

**Sb $t0,-3($s1)**

**优化后共执行1501条指令。**

1. 请编写一个名为fib(n)的递归函数来计算第n个斐波那契数。斐波那契数列的计算公式为：f(n+2)=f(n+1)+f(n)，f(1)=f(2)=1。
2. 用MARS模拟器测试程序是否正确。
3. 如果栈空间容量为4KB，请估算递归调用次数的极限。

解：

**Addi $s2,$0,n**

**Addi $t0,$0,1**

**Addi $t1,$0,1**

**Loop:**

**Add $t2,$t0,$t1**

**Add $t0,$t1,$0**

**Add $t1,$t2,$0**

**Addi $s0,$s0,1**

**Blt $s0,$s1,loop**

2:

**设函数入栈的寄存器个数为n，则栈容量需求为4n个字节，**

**故极限为： 4 / 4n K = (1 /n) K**

5.HW5-CPU:

1. 为了将 DM 结果向 ALU 转发，本章采用如图 6-16(a)的思路：从最后一级流水线寄存器

MEM/WB 向 ALU 转发。但这个设计在执行如下指令序列时必须暂停一个时钟周期。设

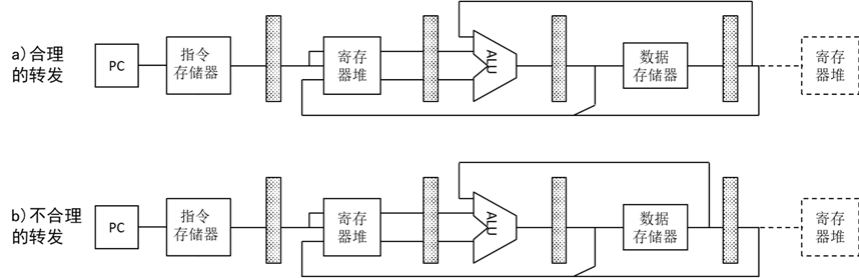
计师认为采用如图 6-16(b)的思路就可以解决这个问题：从直接从 DM 向 ALU 转发。虽

然图 6-16(b)可以解决上述问题，但却使得流水线时钟频率下降了，请分析具体原因

（假设 IM 读出、RF 读出、ALU、DM 读出的延迟均为 L）。

lw $1, xxx

add yyy, $1, zzz



**解：DM和ALU相连使得延迟从L变为了2L，最大时钟频率变成了1/2L,变慢了1倍。**

2. 在流水中插入寄存器可以提高时钟频率。但是，随着级数增多，流水线性能提升会遇

到瓶颈。首先，寄存器自身的时序开销（如寄存器建立时间与保持时间等）对于性能

改善的影响越来越大。其次，随着流水线级数的增长，分支冒险会导致流水线排空的

问题愈发严重。第三，数据冒险造成的暂停也会随之增多。

假设：5 级流水线 CPI 为 1.2，且每增加 1 级，CPI 增加 0.1；单周期 CPU 关键路径延迟

为 800ps，寄存器自身时序开销为 50ps。

1) 建立 CPI 与流水线级数 N（N≥5）的计算公式。

2) 建立时钟周期延迟 Tc 与流水线级数 N 的计算公式。

3) 给出一条指令执行时间的计算公式。

4) 请指出 N 为多少时，流水线性能最好。

5) 请指出 N 为多少时，流水线性能改善最为显著

解：  
**1：**

**2:**

**3:一条指令执行时间为CPI \* Tc为:**

**4:易得，当N = 10时T取最小值，性能最好**

**5:当N = 6时T的变化最大，改善最为显著。**

3. 设计师将单周期数据通路改造为如图 6-17 所示的 3 级流水线。假设 RF 不支持内部转

发

1) 流水线是否仍然可能会因为 beq 而需要清空流水线？如果会，最多有几条指令会

被清除？

2) 假设指令集只有{lw，add}，请以 rs 寄存器为例，增加旁路以应对所有的数据冒险

可能。给出思路即可，不用讨论因此带来的 MUX 及其控制。

3) 对于第 2 问的指令集，能否消除 rs 寄存器相关的全部数据冒险？为什么？

解：

**1:仍然会，最多会有1条指令被清除。**

**2:在第二级流水线寄存器和DM处提供转发数据给ALU的RS位计算。**

**3:能，因为只有这两处地方涉及到RS数据的更改。**

4. 同样是从 DM 向 ALU 的转发，图 6-18(a)的设计会使得 5 级流水线性能下降。请分析图

6-18(b)的设计会导致 3 级流水线性能下降吗？假设 IM 读出、RF 读出、ALU、DM 读出

的延迟均为 L，忽略所有控制器延迟及 MUX 延迟。

**解：不会下降，中间的延迟最坏为2L，而如果像6-18(B)那样，延迟最坏也只会是2L，故性能不会下降。**

5. 下图所示的流水线 CPU 执行如下指令序列。

lw $1, 0($2)

addi $1, $1, $1

sw $1, 0($2)

lw $1, 4($2)

sw $1, 8($2)

1) 分析上述指令执行过程中流水线共计需要暂停多少个时钟周期。

2) 是否可以增加转发来提升流水线性能？如果可以，请简述设计思路。

解：

**1： lw-addi 暂停1个周期，addi-sw暂停1个周期，lw-sw暂停1个周期，总共暂停3个周期。**

**2： 可以，从W级转发数据给M级的数据写入端，在M端，将ALU的计算结果转发给E端。**

6. 在第 5 题图中所示的流水线 CPU 执行某程序，其指令分布如下：load 占 15%，store 为

10%，分支指令为 10%，R 型计算类指令为 65%。假设：load-R 导致暂停概率为 30%；

load-store 导致的暂停概率为 5%；分支指令预测成功率为 75%。计算流水线执行该程

序的 CPI。

解：

**1.3 \* 30% + 1.1 \* 5% + 1 \* 10% + 1.25 \* 10% + 1 \* 65% = 1.32**

7. 如图 6-19 所示，某 MIPS 标准 5 级流水线仅支持 M 级向 D 级的转发（注意：寄存器堆

无内部转发）。某程序员编写了如下 MIPS 代码，请回答下列问题。

1) 请指出上述指令片段在上述流水线中执行时存在的所有数据相关。

2) 请通过调整指令顺序来优化上述指令片段以最大化减少暂停。

3) 对于优化前和优化后指令片段，分别给出流水线的执行时间，并说明理由。示例：

对于 2 条无冒险的指令片段，则流水线执行时间为 6 个时钟周期。

解：

**1：lw-sw 有关于$1的两次数据相关，lw-sub关于$s1相关，add-sub关于$s3相关。**

**2：**

**lw $1,0($2)**

**add $3,$s2,$2**

**or $5,$5,$6**

**sw $1,0($1)**

**sub $3,$2,$2**

**3：**

**对于优化后的指令片段，执行时间为9个时钟周期**

**对于优化前的指令片段，执行时间为13个时钟周期**

6.HW6-存储:

1. 一标准流水线 CPU 的理想 CPI 为 1，其执行的程序片段中 load 和 store 类指令占 30%， 其余指令均为 R 型指令。主存延迟为 50 个时钟周期。L1 级 cache 性能为：命中时间为 1 个时钟周期，缺失率为 2%。L2 级 cache 性能为：命中时间为 10 个时钟周期，缺失率 为 5%。针对如下情况分别计算流水线的实际 CPI：没有 cache；只有 L1 级 cache；L2 级 cache。

**解：当没有cache时：**

**CPI = 1 + 50 + 50 \* 30% = 65 + 1 = 66**

**当只有L1级cache时：**

**CPI = 1 + 50 \* 2 % + 40 \* 2 % \* 30 % = 1.3 + 1 = 2.3**

**当有L2级cache时：**

**CPI =1 + （2 % + 0.6%） \* （10 + 5% \* 50） = 1 + 0.325 = 1.325**

2. 直接映射 cache 参数如下：cache 的数据容量为 16KB，cache 块为 16B。现该 cache 接

入一个 32 位 cpu。请给出 cache 的 TIO 结构以及包含标记后的 cache 块容量。

解：

**Cache中共有 16KB / 16B = 2^10 块**

**故Offset共有 位**

**Index共有 位**

**Tag共有 32 - 4 -10 = 18 位**

**包含标记后的cache块容量为16B + 18b + 1b = 16B + 19b**

3. 组相联映射 cache 参数如下：cache 的数据容量为 512KB，16 路相联，cache 块为 32B。

现该 cache 接入一个 32 位 cpu。请给出 cache 的 TIO 结构。

**解：cache中共有 512KB / 32B = 2^14块共有 2^14 / 16 = 2 ^ 10组**

**故Offset共有位**

**Index共有位**

**Tag共有32 - 10 - 5 = 17位**

4. 对于第 3 问的系统，cache 采用写回策略，cache 命中时间为 1 个时钟周期，从主存调

入一块或回写一块的代价均为 100 个时钟周期。某程序片段将起始地址为 0000\_0000h

的 1MB 内存单元全部初始化为 0，代码如下所示。

int \*p=0x0 ;

for ( int i=0; i<1024\*1024/4; i++ )

\*p++ = 0 ;

1) 计算该程序片段中的数据访问缺失率（忽略指令缺失等因素）。

2) 计算该程序片段的数据访问实际需要多少个时钟周期。

3) 计算程序片段的存储访问次数与实际时钟周期数之比。

解：

**1：数据访问缺失率为：1/8 = 12.5%**

**2：对于1个块来说，总共时间为100 + 7 = 107 个时钟周期**

**总共有32K块，故时间为 107 \* 32K**

**由于需要替换，故还需要100 \* 16K**

**总时间为107 \* 32K + 100 \* 16K = 5024K**

**3:**

**程序总访问次数为 1MB / 4 = 256 K**

**访问次数:实际时钟周期数约为1：20**

5. 假设某系统的虚页和物理页尺寸均为 8KB，40 位虚地址，物理主存容量为 32GB。请问

虚页号与物理页号的位数分别是多少。

**解：页内位移为:位**

**虚页号位数：40 - 13 = 27 位**

**物理页位数：位**

6. 设计师给第 5 问的系统设计了 TLB。TLB 采用 2 路组相联结构，共有 256 个页表项。TLB

中的每个页表项结构如下图所示

1) 请问每个页表项的位数是多少。

2) 页表的总存储容量是多少位？

3) 操作系统设计团队希望将页面尺寸从 8KB 降低至 4KB，但硬件设计团队认为会增

加硬件开销，因此不同意改变页面尺寸。你作为硬件设计师团队负责人，请陈述理

由。

**解:**

**1:位数为：**

**2:总存储容量位数为：40 -13 - 7 = 20 位**

**3:页表项位数会增加到46位**

7. 现有 4K × 8 位容量的 DRAM 存储芯片。

1) DRAM 芯片内置译码器输出的行/列选择线各是多少条。

2) DRAM 芯片内置的刷新地址计数器位数为多少位。

3) DRAM 每刷新周期刷新一行存储单元是多少位。

4) 主存容量为 8K × 8 位，需使用 DRAM 芯片的数量。

5) 给出每个 DRAM 芯片的片选控制信号的逻辑表达式。

**解：**

**1：行选择线有:条，列选择线有:条**

**2: 刷新地址计数器位数为6位。**

**3：为512位。**

**4：需要 8K \* 8 / (4K \* 8) = 2 个芯片**

**5：地址位A12[12:0] A12为1则选中第二个芯片，A12为0则选择第一个芯片。**

**A12[11:0]为段选信号。**