CPE/EE 421
Microcomputers

WEEK #10

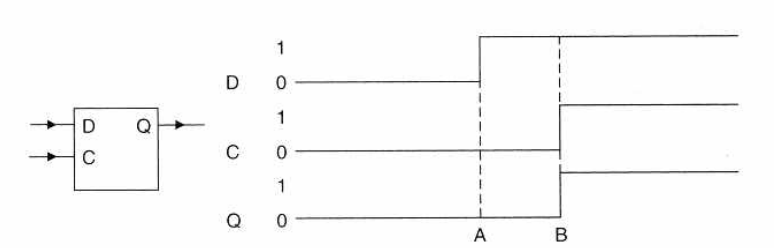
U A H

Interpreting the Timing Diagram

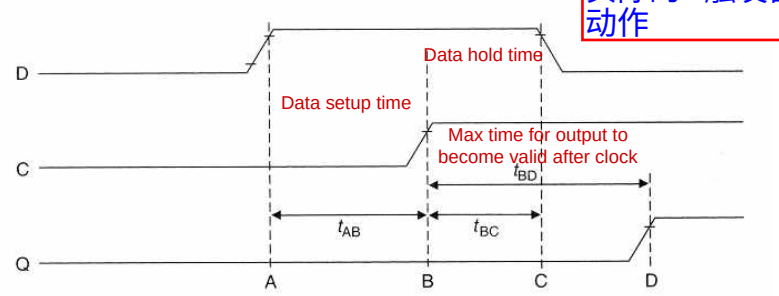
The 68000 Read Cycle

Timing Diagram of a Simple Flip-Flop

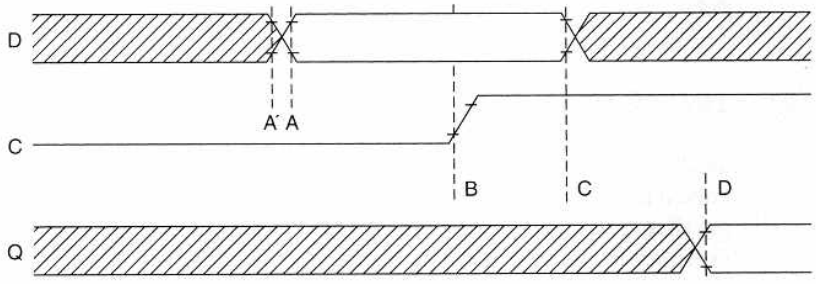
Idealized form of the timing diagram 理想形式的时序图



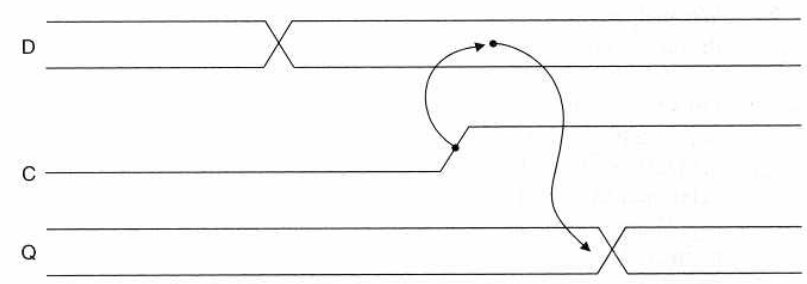
Actual behavior of a D flip-flop 实际的D触发器的动作



General form of the timing diagram 通常形式的时序图



An alternative form of the timing diagram 另一种形式的时序图

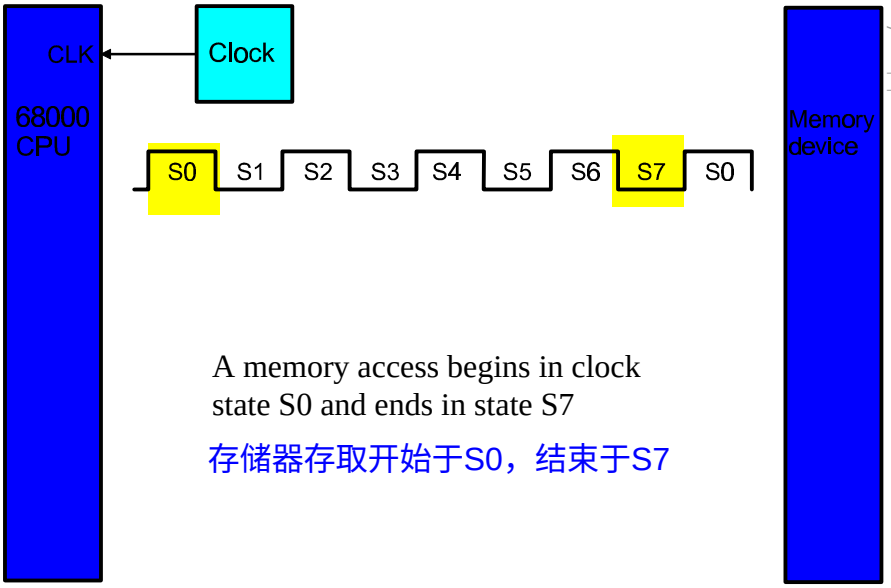


The Clock

- A microprocessor requires a clock that provides a stream of timing pulses to control its internal operations
- A 68000 memory access takes a minimum of eight clock states numbered from clock state S0 to clock state S7

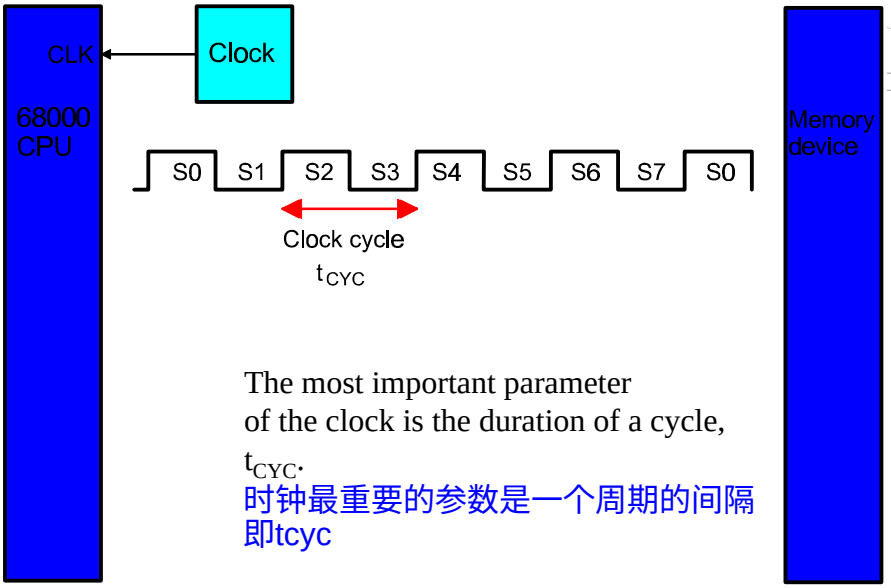
微处理器需要一个时钟来提供控制其内部操作的时间脉冲流

68000存储器存取采用最少8种时钟状态，标号为状态S0到S7



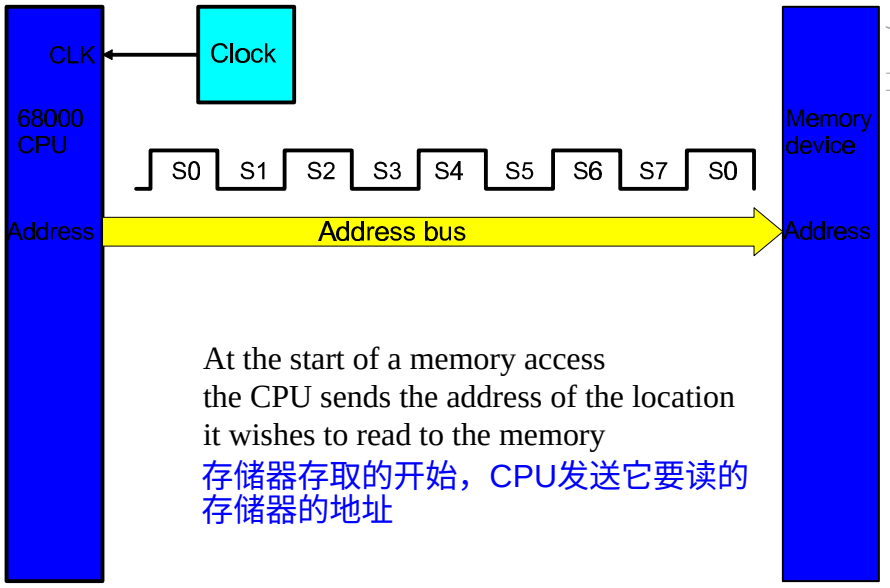
A memory access begins in clock state S0 and ends in state S7

存储器存取开始于S0，结束于S7



The most important parameter of the clock is the duration of a cycle, t_{CYC}.

时钟最重要的参数是一个周期的间隔即tcyc

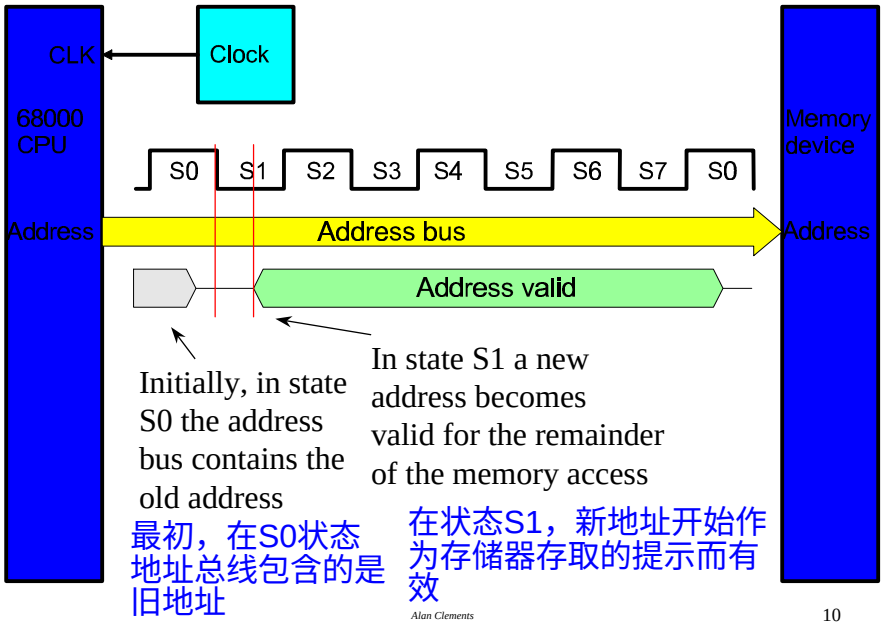


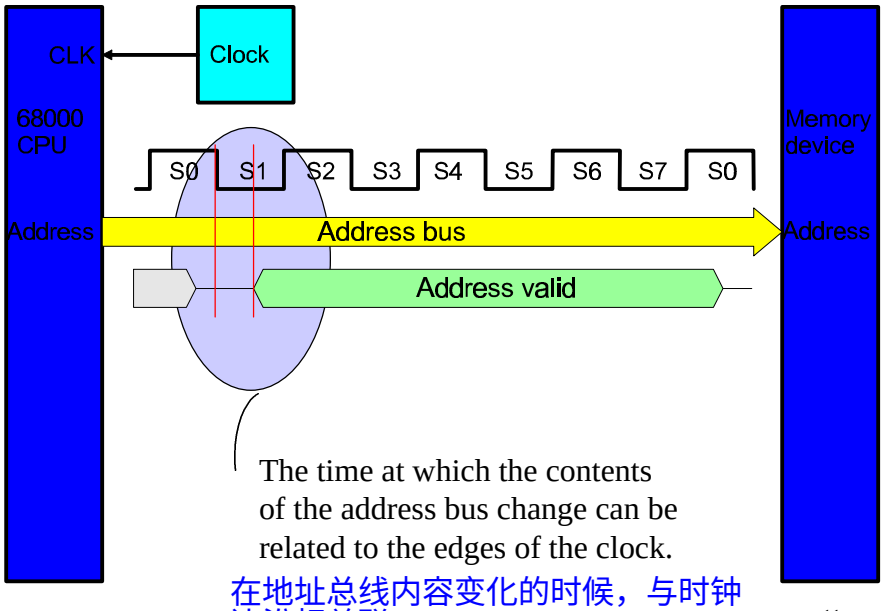
At the start of a memory access the CPU sends the address of the location it wishes to read to the memory

存储器存取的开始，CPU发送它要读的存储器的地址

Address Timing 时钟

- We are interested in when the 68000 generates a new address for use in the current memory access
  我们对68000什么时候形成使用当前存储器存取的地址很感兴趣
- The next slide shows the relationship between the new address and the state of the 68000's clock
  下一步表明了新地址和68000的时钟状态的关系

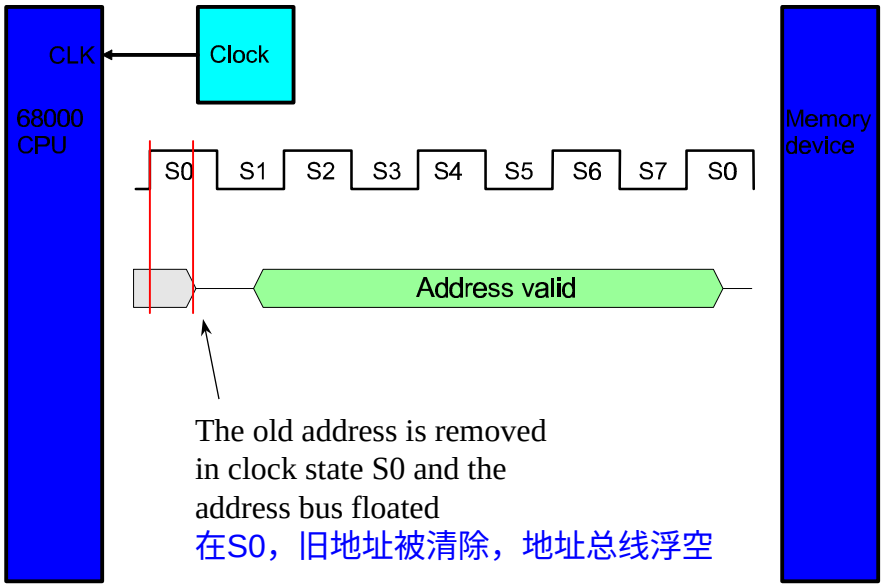




在地址总线内容变化的时候，与时钟边沿相关联

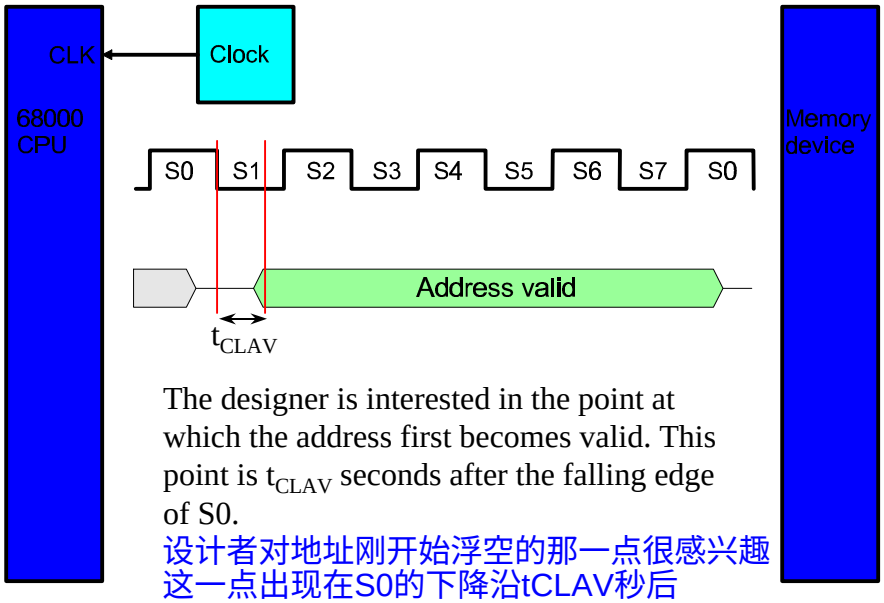
Address Timing

- Let's look at the sequence of events that govern the timing of the address bus 我们看一系列控制地址总线时序的实例
- The "old" address is removed in state S0 在S0状态，旧地址被清除
- The address bus is floated for a short time, and the CPU puts out a new address in state S1 在S1状态，地址总线浮空一段时间，CPU输出新地址



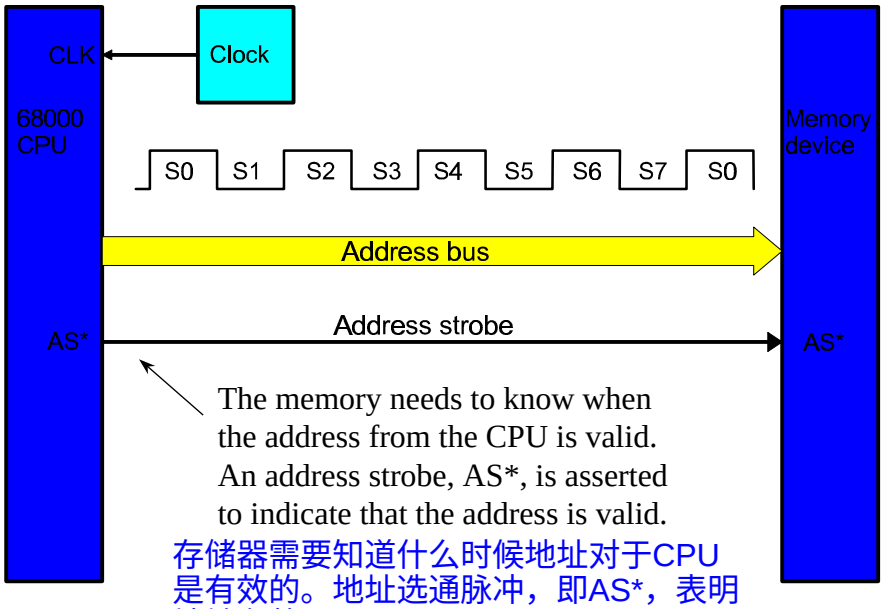
The old address is removed in clock state S0 and the address bus floated

在S0，旧地址被清除，地址总线浮空



The designer is interested in the point at which the address first becomes valid. This point is t_{CLAV} seconds after the falling edge of S0.

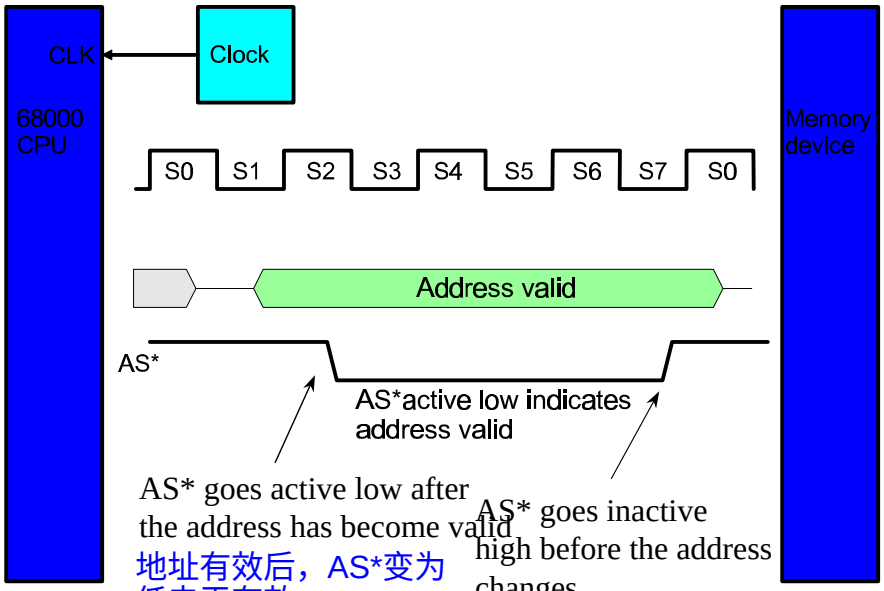
设计者对地址刚开始浮空的那一点很感兴趣
这一点出现在S0的下降沿tCLAV秒后



存储器需要知道什么时候地址对于CPU是有效的。地址选通脉冲，即AS*，表明地址有效

Address and Address Strobe 地址和地址选通脉冲

- We are interested in the relationship between the time at which the address is valid and the time at which the address strobe, AS*, is asserted
- When AS* is *active-low* it indicates that the address is valid 当AS*由高变低，表明地址有效
- We now look at the timing of the clock, the address, and the address strobe

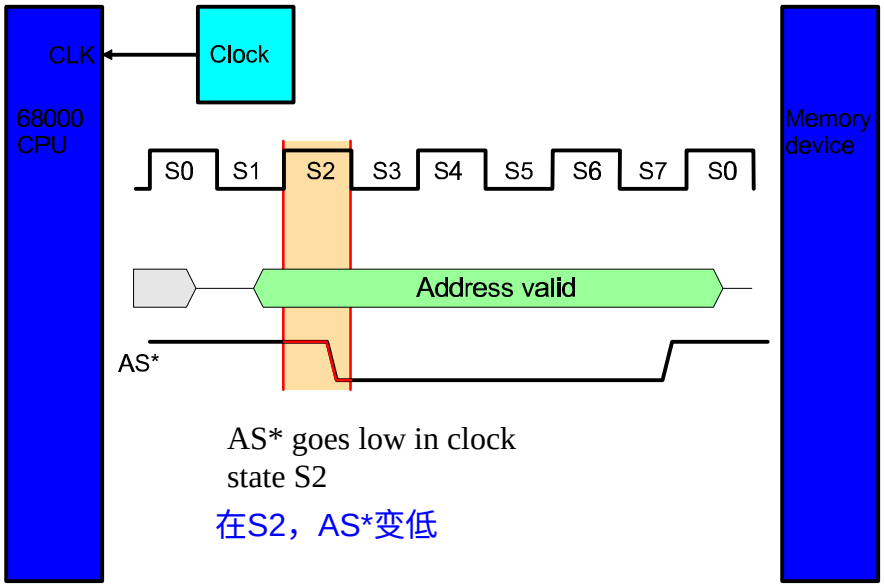


AS* goes active low after the address has become valid

地址有效后，AS*变为低电平有效

AS* goes inactive high before the address changes

在地址变化前，AS*变为高电平无效

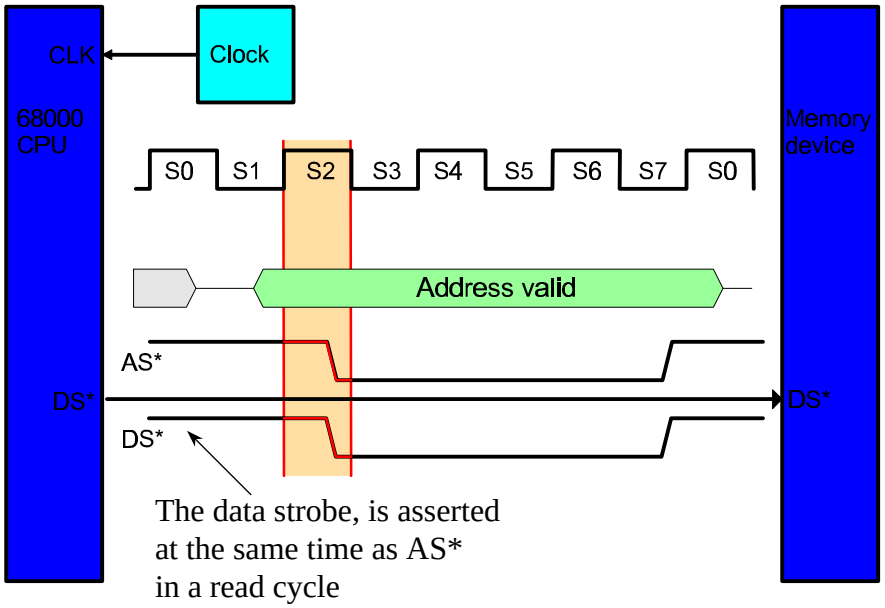


AS* goes low in clock state S2

在S2，AS*变低

The Data Strobes 数据选通

- The 68000 has two data strobes LDS* and UDS*. These select the lower byte or the upper byte of a word during a memory access
  68000有两个数据选通信号LDS*和UDS*。在存取期间，他们用来选择低/高字节
- To keep things simple, we will use a single data strobe, DS*
  简化起见，我们用一个数据选通信号DS*
- The timing of DS* in a read cycle is the same as the address strobe, AS*
  在读周期，DS*和AS*时序一样



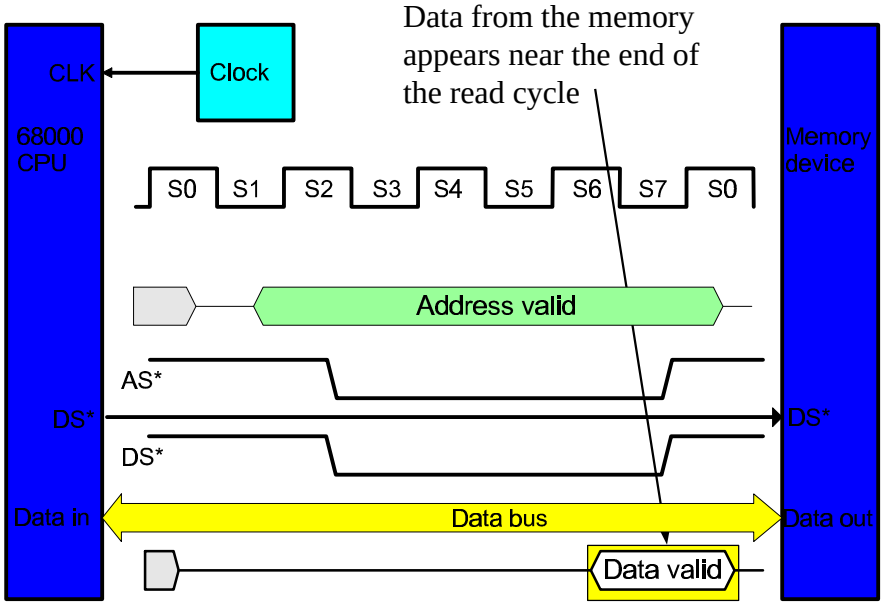
The Data Bus

- During a read cycle the memory provides the CPU with data
- The next slide shows the data bus and the timing of the data signal
- Note that valid data does not appear on the data bus until near the end of the read cycle

在读周期，存储器为CPU提供数据

下一步显示了数据总线和数据信号的时序

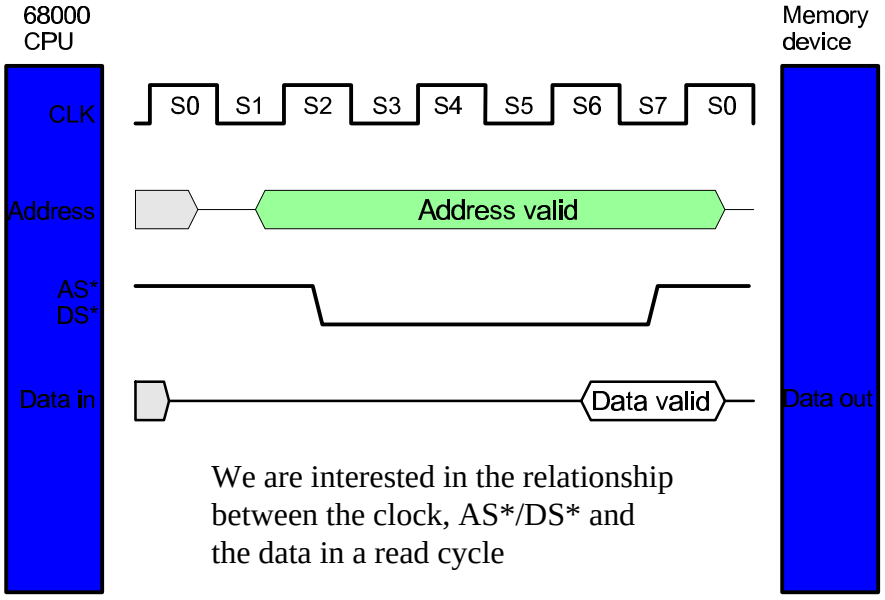
注意有效信号直到接近读周期的末尾才出现在数据总线上



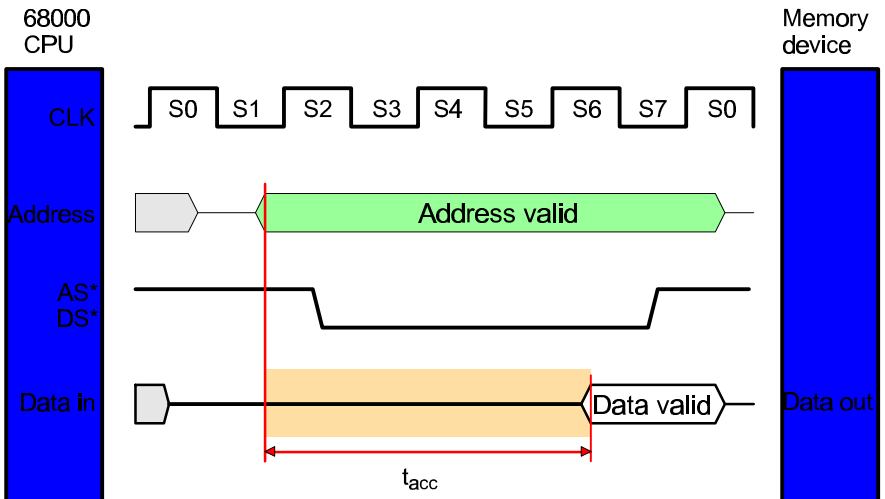
Analyzing the Timing Diagram 分析时序图

- We are going to redraw the timing diagram to remove clutter凌乱
- We aren't interested in the signal paths themselves, only in the relationship between the signals

我们对信路本身不感兴趣，只对信号间的关系感兴趣

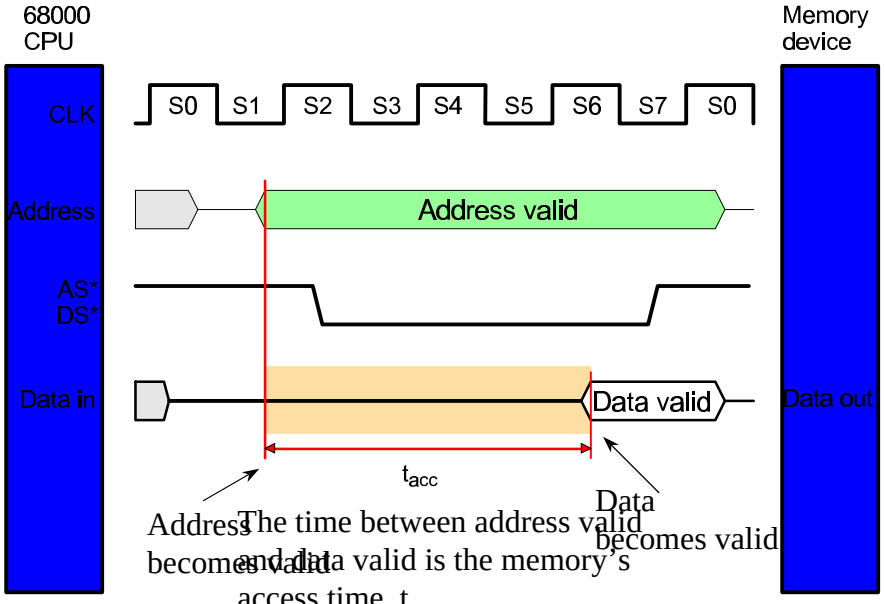


We are interested in the relationship between the clock, AS*/DS* and the data in a read cycle



The earliest time at which the memory can begin to access data is measured from the point at which the address is first valid

存储器最初开始接受数据到地址开始有效的时间测量值为tacc

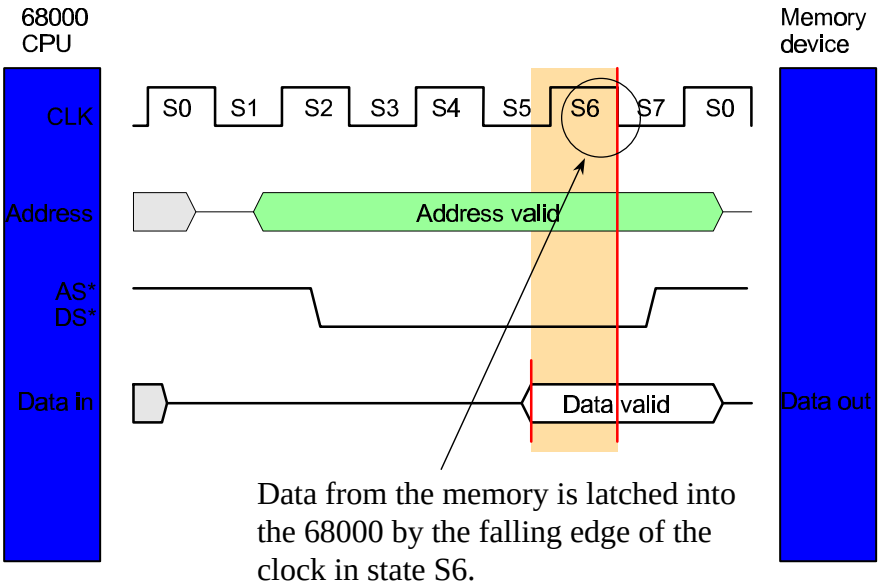


Calculating the Access Time

- We need to calculate the memory's access time
- By knowing the access time, we can use the appropriate memory component
- Equally, if we select a given memory component, we can calculate whether its access time is adequate for a particular system

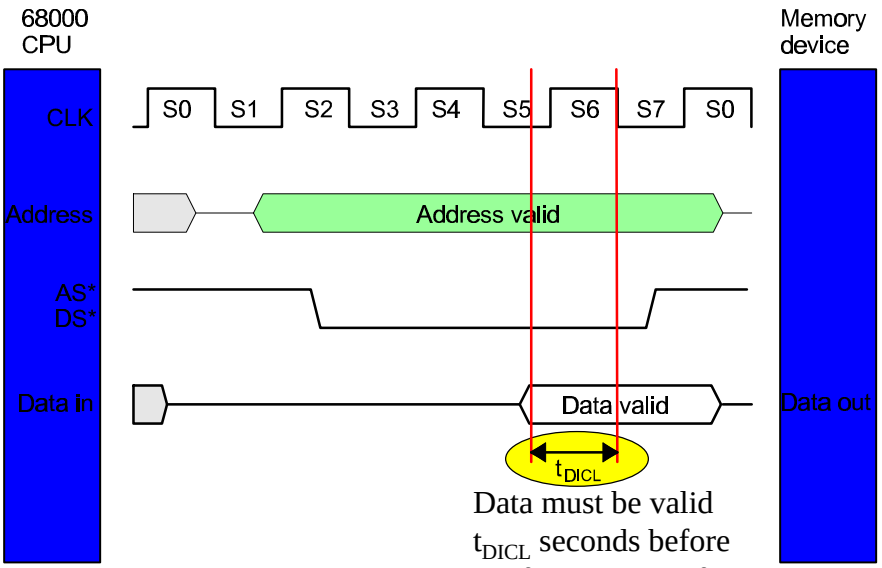
知道了接受时间，我们就能使用最优的存储器件

一般地，如果我们选择了一个存储器件，我们能算出它的接受时间是否适合特定系统



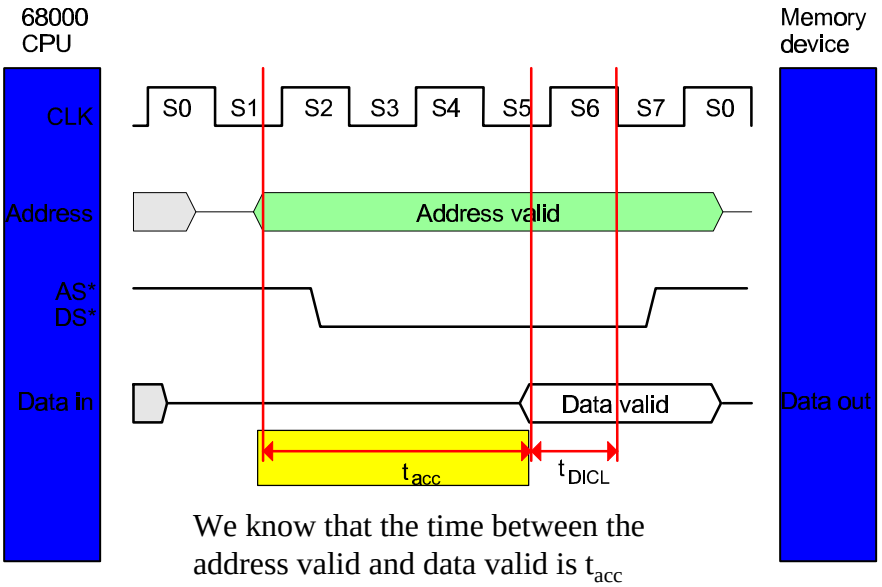
Data from the memory is latched into the 68000 by the falling edge of the clock in state S6.

在S6的下降沿，来自存储器的数据被锁存进68000



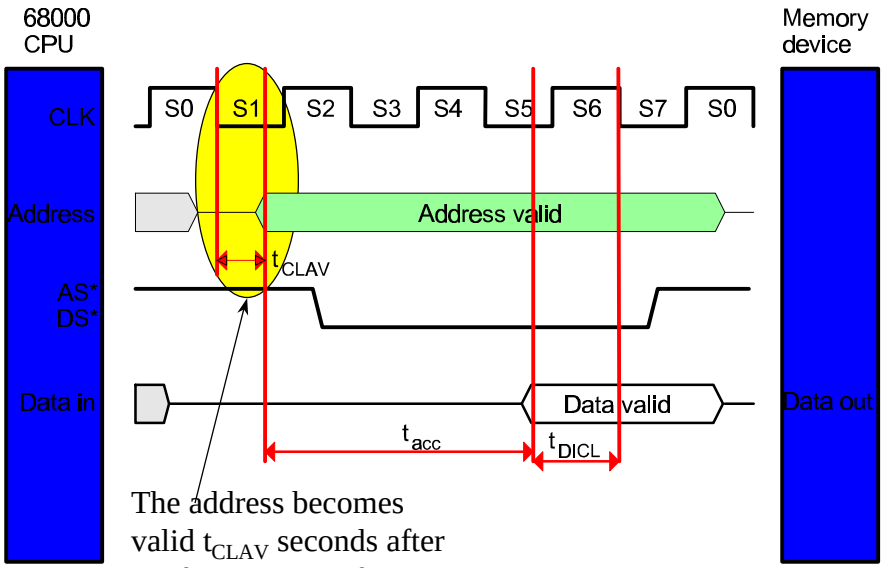
Data must be valid t_{DICL} seconds before the falling edge of S6

在S6的下降沿之前
数据必须有效tDICL秒



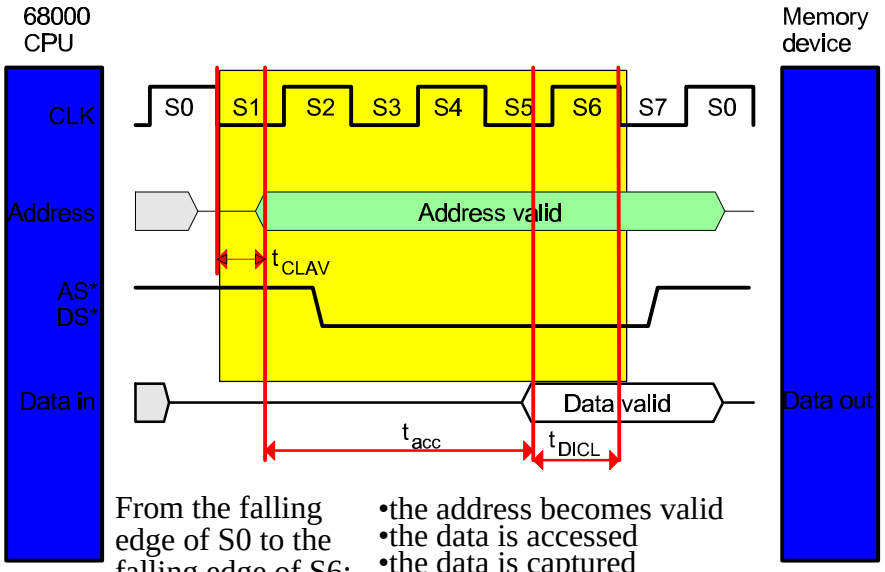
We know that the time between the address valid and data valid is t_{acc}

我们知道地址有效到数据有效的时间是tacc



The address becomes valid t_{CLAV} seconds after the falling edge of S0

在S0的下降沿后的tCLAV秒后，地址有效

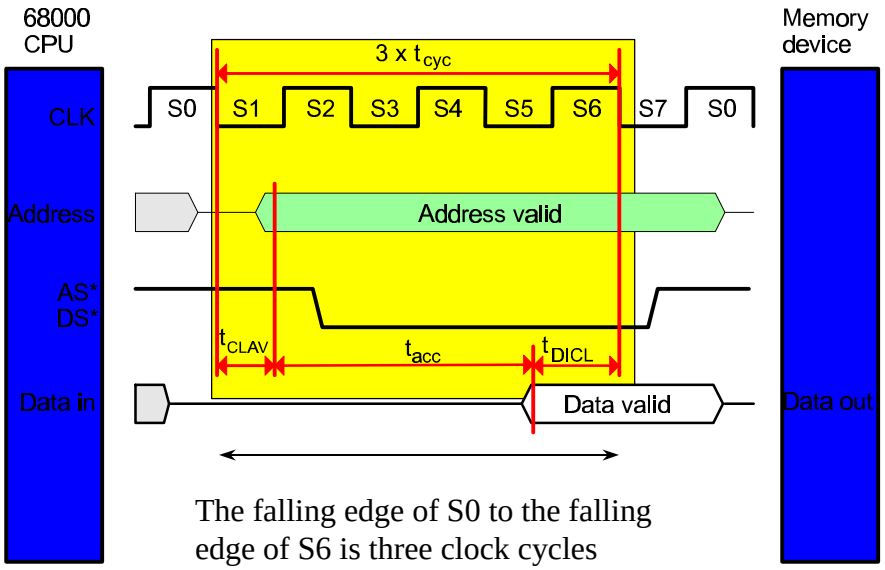


From the falling edge of S0 to the falling edge of S6:

- the address becomes valid
- the data is accessed
- the data is captured

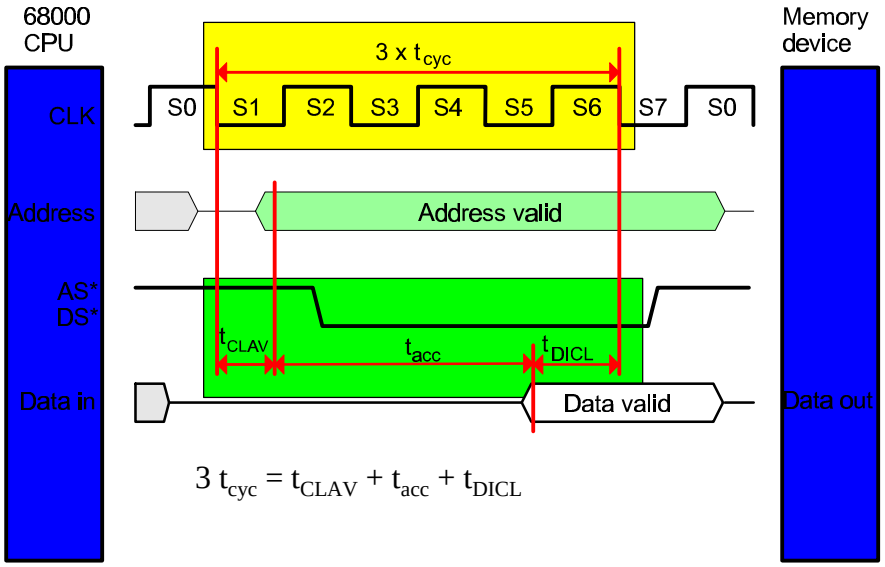
从S0的下降沿到S6的下降沿

- 地址变有效
- 数据被接受
- 数据被捕获



The falling edge of S0 to the falling edge of S6 is three clock cycles

在S0 下降沿与S6的下降沿间有3个时钟周期



$3\ t_{cyc} = t_{CLAV} + t_{acc} + t_{DICL}$

Timing Example

- 68000 clock 8 MHz t_{CYC} = 125 ns
- 68000 CPU t_{CLAV} = 70 ns
- 68000 CPU t_{DICL} = 15 ns
- What is the minimum t_{acc}?
- 3 $t_{CYC} = t_{CLAV} + t_{acc} + t_{DICL}$
- 375 = 70 + t_{acc} + 15
- t_{acc} = 290 ns