模拟与数字电路

Analog and Digital Circuits

课程主页 扫一扫

第 十三讲： **寄存器与存储器**

Lecture 13: **Register Files and Memories**

主　　讲： 陈 迟 晓

Instructor : Chixiao Chen

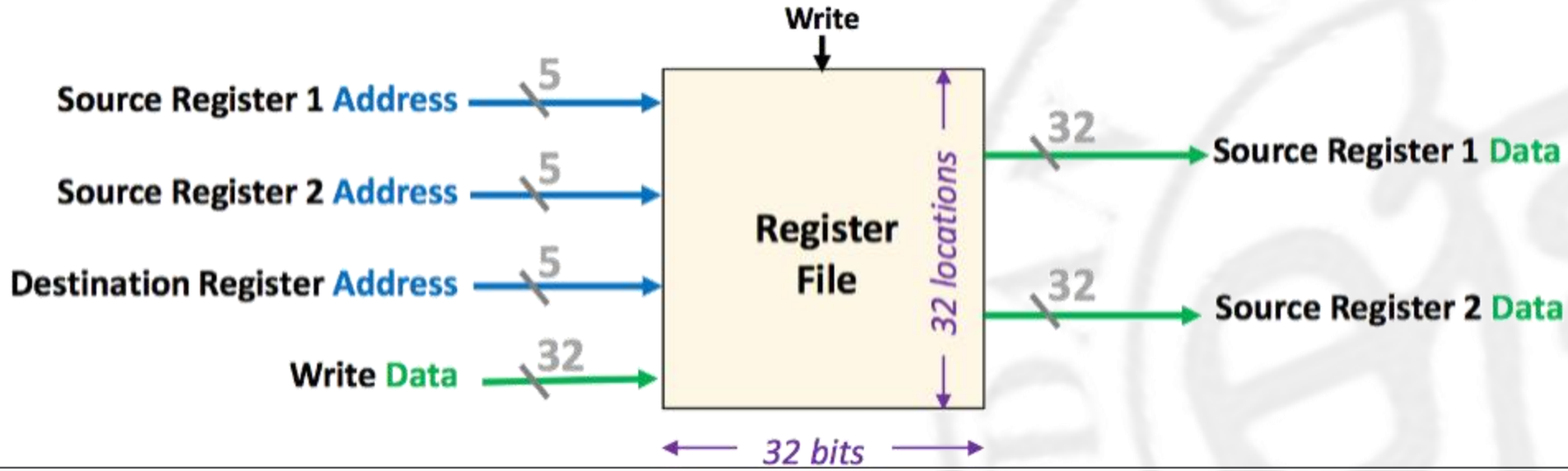
提纲

- 复习

- 寄存器、寄存器组、寄存器列表
- 存储器电路与结构

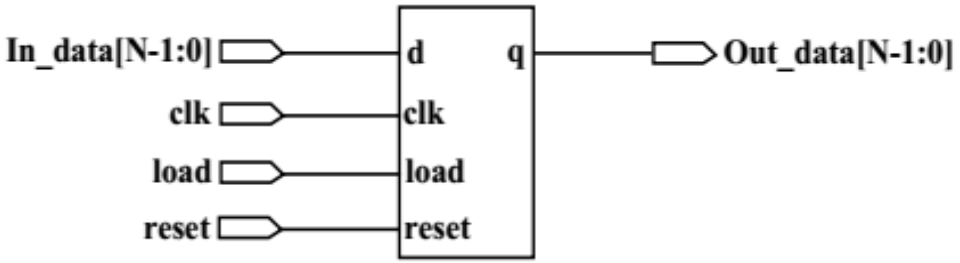
寄存器列表 Register File

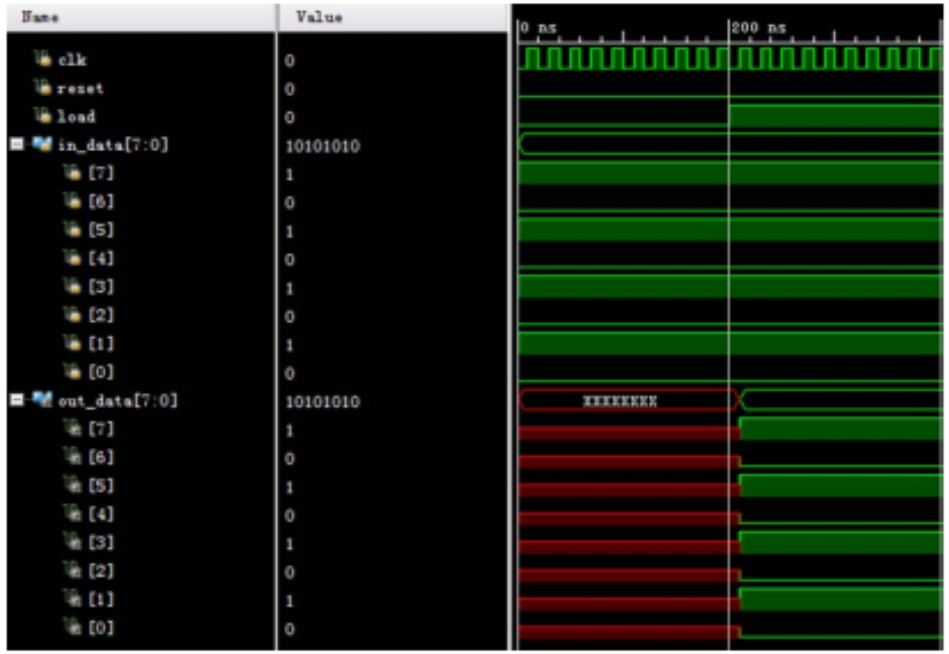
- 寄存器是RISC (load-store 架构)的核心单元
  - ALU 计算的输入、输出仅能是寄存器列表



N位寄存器的功能描述

- 设计一个N位寄存器，它可以在需要时从输入线in_data加载一个值，我们给D触发器增加一根输入线load，当我们想要从in_data加载一个值时，就把load设置为1，那么在下一个时钟上升沿，in_data的值将被存储在q中。





N位寄存器的Verilog功能

```
module reg_N
 #(parameter N = 8)
   (
      input  clk,
      input  reset,
      input [N-1:0] in_data,
      input   load,
      output  reg [N-1:0] out_data
   );
always @(posedge clk, posedge reset)
      if(reset)
            out_data <= 0;
      else if(load == 1)
            out_data <= in_data;
   endmodule
```

- 如果我们想修改寄存器的位宽，可以使用Verilog的实例化语句。我们可以实现一个如下所示的16位寄存器，称为fReg。

```
reg_N #(
       .N(16))
      fReg(.clk(clk),
      .reset(reset),
      .load(load),
      .in_data(indata),
      .out_data(out_data)
);
```

寄存器组

- 寄存器组是由一组拥有同一个输入端口和一个或多个输出端口的寄存器组成。写地址信号w_addr指定了数据存储位置，读取地址信号r_addr指定数据检索位置。
- 常用于快速、临时存储。

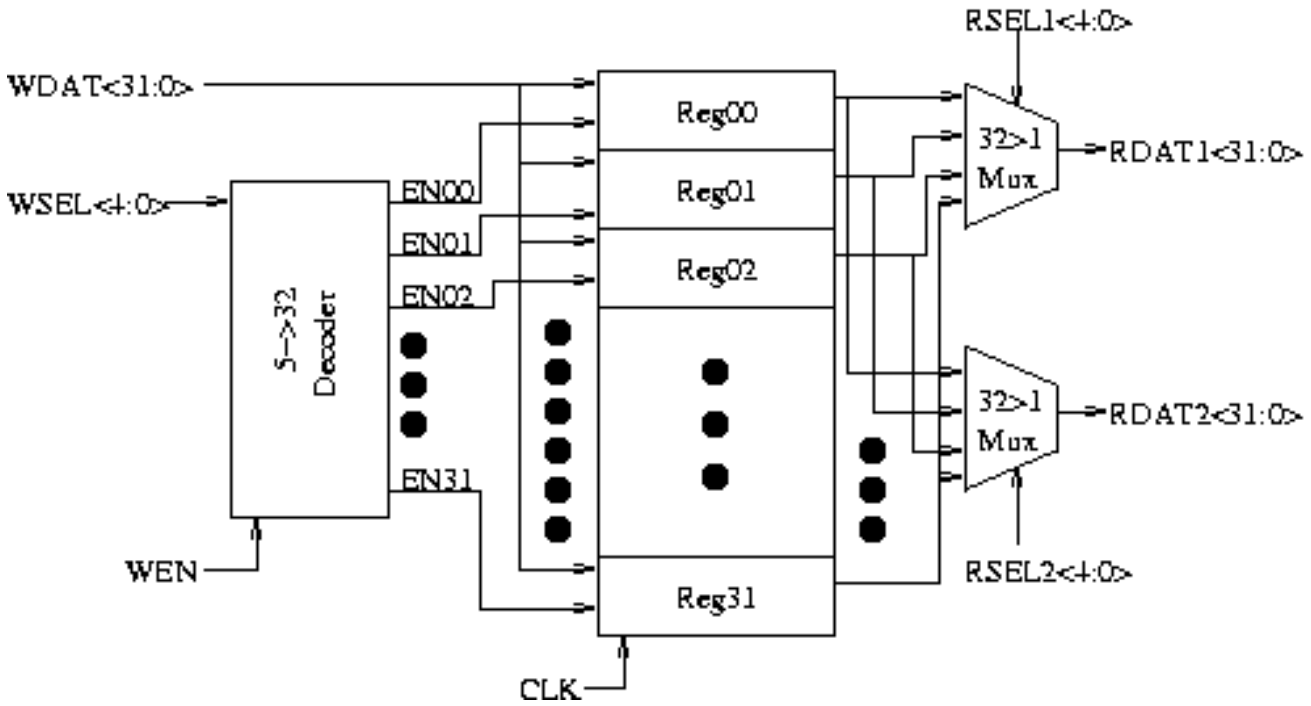
```
module reg_file
  #(
parameter N = 8，//比特数
W = 2) //地址比特数
      (
        input  clk,
        input  wr_en,
        input [W-1:0] w_addr,r_addr,
        input [BN-1:0]  w_data,
        output  [BN-1:0]  r_data
    );
     reg [BN-1:0] array_reg[2**W-1:0];
```

```
always @(posedge clk)
      if(wr_en)
        array_reg[w_addr] <= w_data;
        assign r_data = array_reg[r_addr];
  endmodule
```

一个信号被用作索引来访问数组中元素

二维数组的数据类型：表示array_reg变量是一个含有[2**W-1:0]个元素的数组，每个元素的数据类型是reg [B-1:0]

寄存器列表的Verilog描述



Register File (2 read ports, 1 write port) Pg.2

```
module regfile(input    clk,
               input    we3,
               input  [4:0]  ra1, ra2, wa3,
               input  [31:0] wd3,
               output [31:0] rd1, rd2);

reg [31:0] rf [31:0];                    //2-D (32x32) array

always @(posedge clk)
  if (we3) rf[wa3] <= wd3;                //write

assign rd1 = (ra1 != 0) ? rf[ra1] : 0;    //read 1
assign rd2 = (ra2 != 0) ? rf[ra2] : 0;    //read 2

Endmodule        // Note: rf[0] is always 0!
```

width

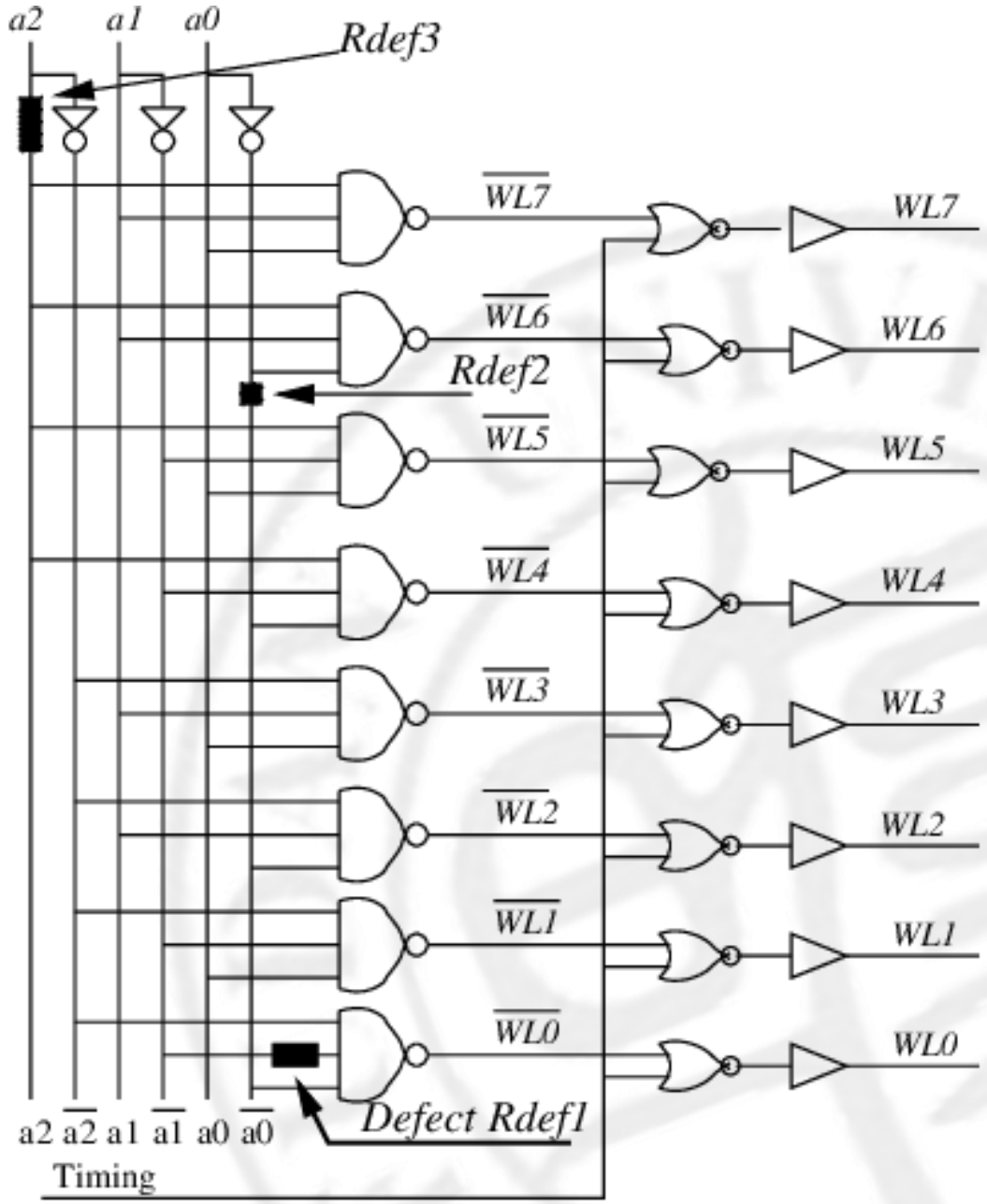
depth

outside of always

地址译码器

- 3-8译码器

```
module Vr74x138a(G1, G2A_L, G2B_L, A, Y_L);
  input G1, G2A_L, G2B_L;
  input [2:0] A;
  output [0:7] Y_L;
  reg [0:7] Y_L;

  always @ (G1 or G2A_L or G2B_L or A) begin
    if (G1 & ~G2A_L & ~G2B_L)
      case (A)
        0: Y_L = 8'b01111111;
        1: Y_L = 8'b10111111;
        2: Y_L = 8'b11011111;
        3: Y_L = 8'b11101111;
        4: Y_L = 8'b11110111;
        5: Y_L = 8'b11111011;
        6: Y_L = 8'b11111101;
        7: Y_L = 8'b11111110;
        default: Y_L = 8'b11111111;
      endcase
    else Y_L = 8'b11111111;
  end
endmodule
```



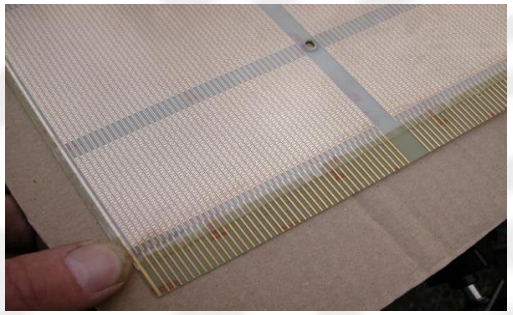
存储器发展历程

- 早期只读存储器
  - 打孔卡片(punched cards)
  - 打孔纸带(punched tape)
  - 电容

Punched cards, From early 1700s through Jaquard Loom, Babbage, and then IBM

Punched paper tape, instruction stream in Harvard Mk 1

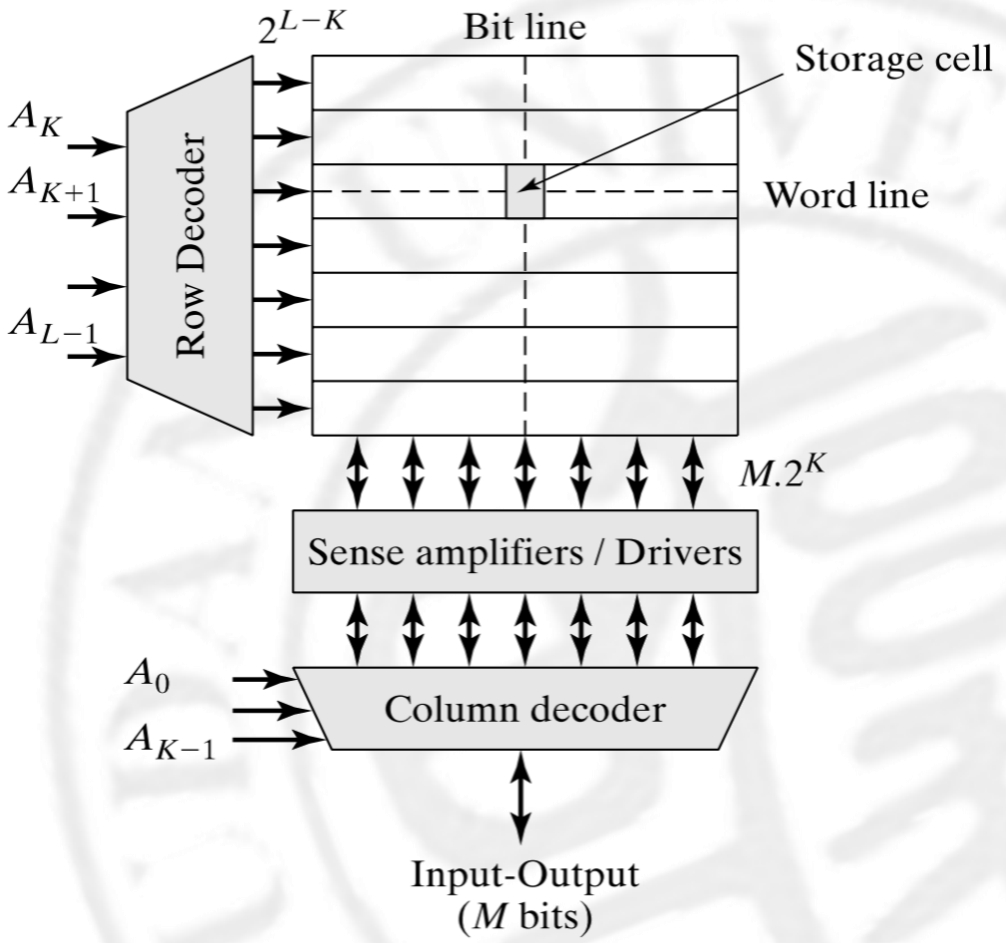
- 半导体存储器
  - 半导体存储器在20世纪70年代开始变得有竞争力
    - 英特尔成立之初旨在开拓半导体存储器市场
    - 早期半导体存储器为静态存储器SRAM
  - 首款商用动态存储器为 Intel 1103
    - 单芯片1Kbit 存储
    - 电容充电维系数据保存



IBM Balanced Capacitor ROS

存储器的通用架构

- Word line 字线 --选通读写行
- Bit line 位线 – 数据输入或者输出
- Row Decoder – 行译码器
- Address Bit 地址位 – 译码器输入
- Sense Amplifier -- 灵敏放大器



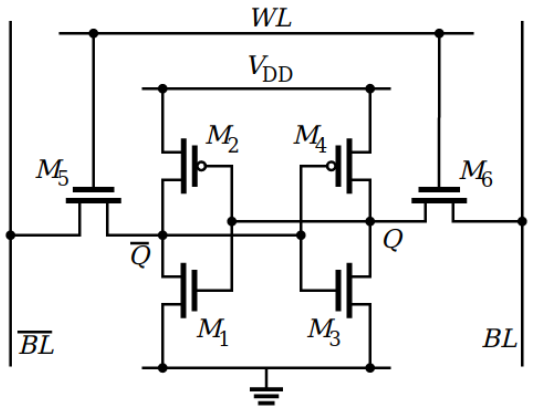
存储器的Verilog实现

与寄存器的差别?

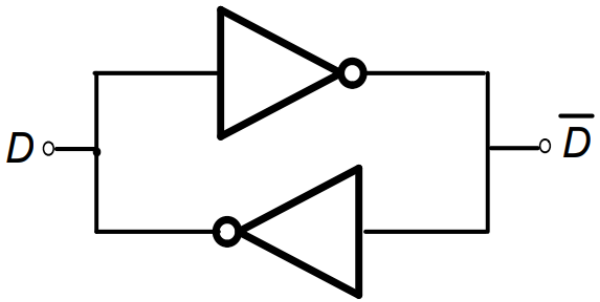
```
module dcmem #(
  parameter MEM_ADDR_WIDTH=5,
  parameter MEM_DATA_WIDTH=16,
  parameter MEM_NUMBER=32
)  (
  input clk,
  input MemWEn,
  input [MEM_ADDR_WIDTH-1:0] addr,
  input [MEM_DATA_WIDTH-1:0] dataw,
  output [MEM_DATA_WIDTH-1:0] datar
);
  reg [MEM_DATA_WIDTH-1:0]RAM [MEM_NUMBER-1:0];

  always@(posedge clk)begin
    if (MemWEn) begin
      RAM[addr]<=dataw;
       end
  end
  assign datar = RAM[addr];
endmodule
```

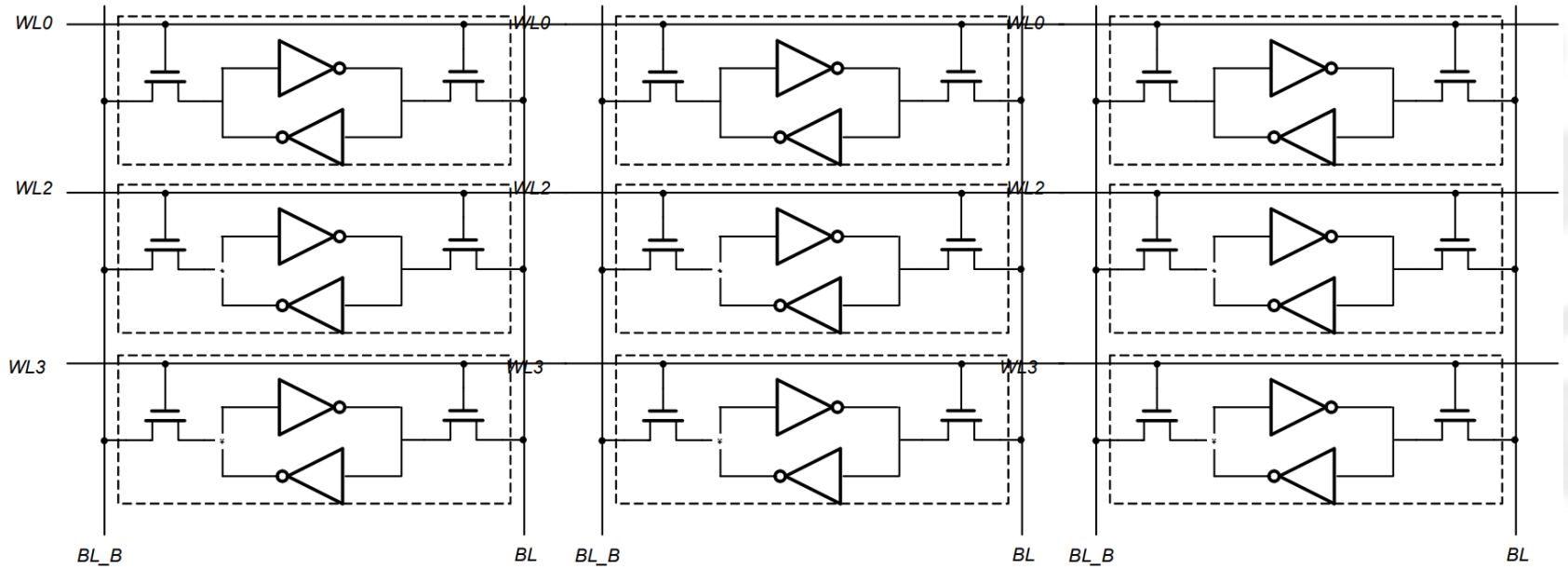
片上存储

- 静态存储器
  - 不需要刷新
  - 噪声裕度大，通常工作在低压
  - 缺点：更大的面积



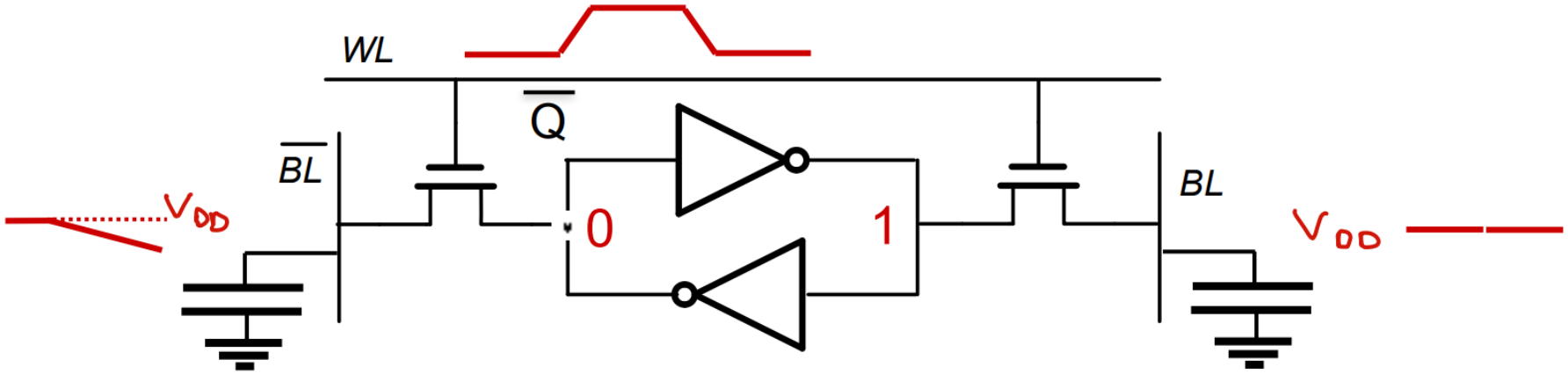
SRAM Memory Cell





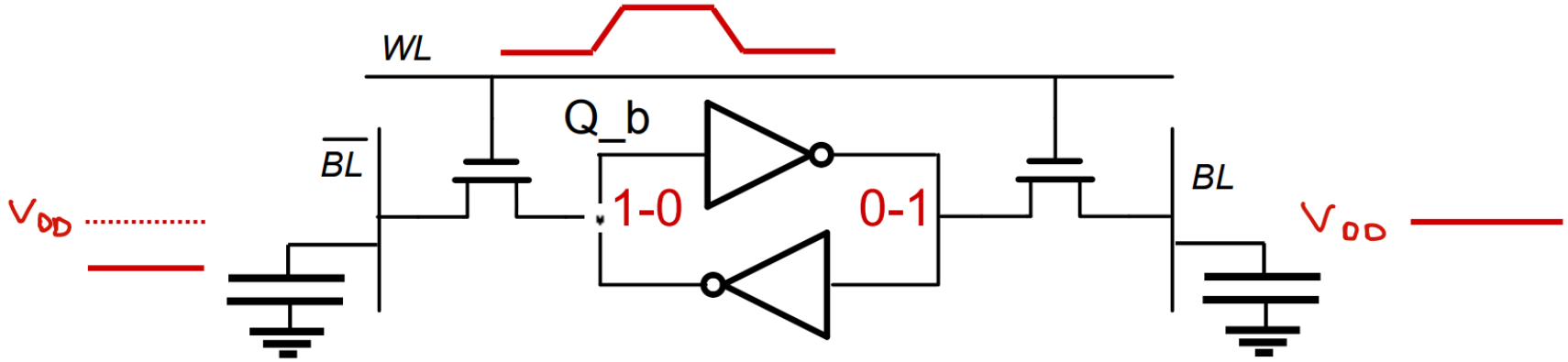
SRAM的读操作

1. Bit lines are “pre-charged” to VDD
2. Word line is driven high (pre-charger is turned off)
3. Cell pulls-down one bit line
4. Differential sensing circuit on periphery is activated to capture value on bit lines.



SRAM的写操作

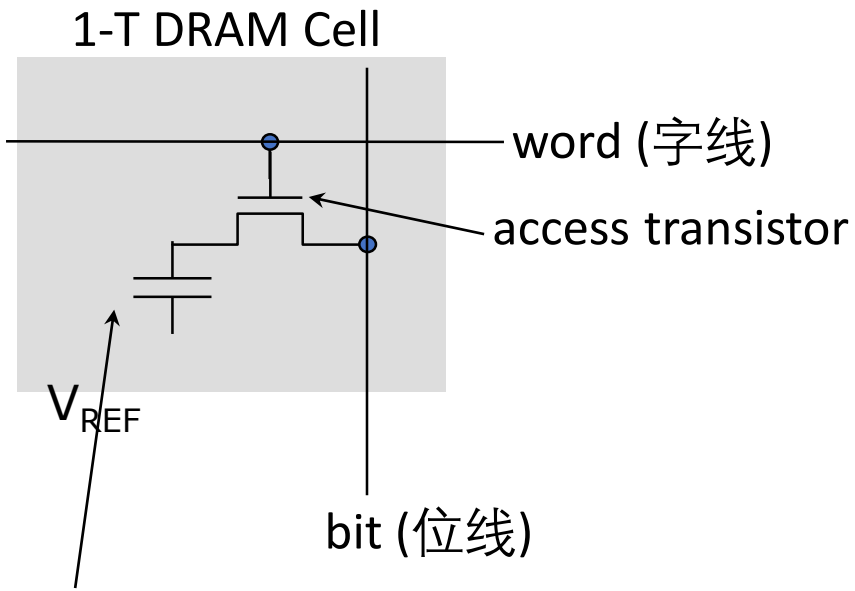
1. Column driver circuit on periphery differentially drives the bit lines
2. Word line is driven high (column driver stays on)
3. One side of cell is driven low, flips the other side

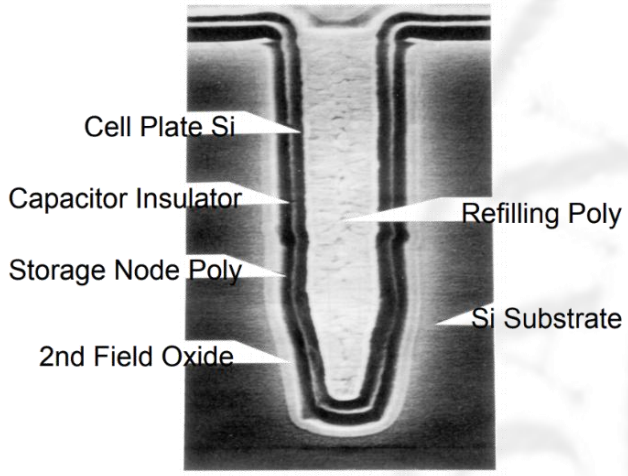


For successful write the access transistor needs to overpower the cell pullup. The transistors are sized to allow this to happen.

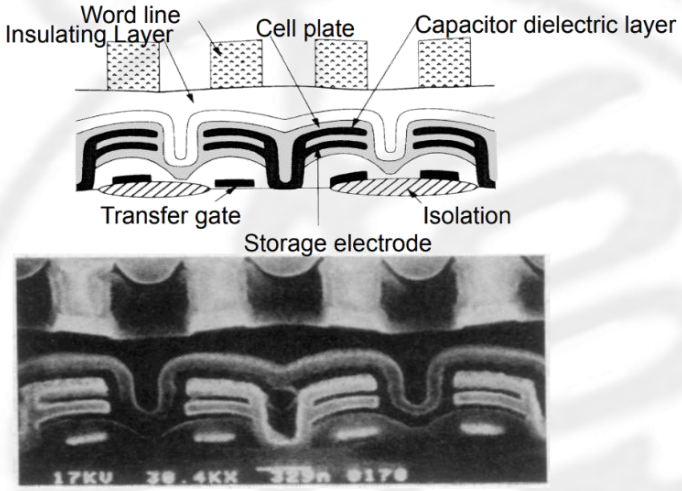
单晶体管动态RAM [Dennard, IBM]

- 用一个电容存储 0/1，用一个开关选择



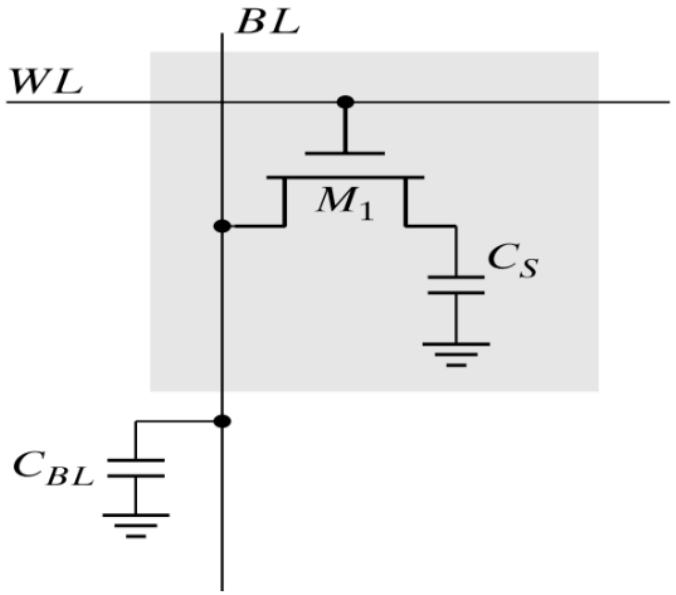


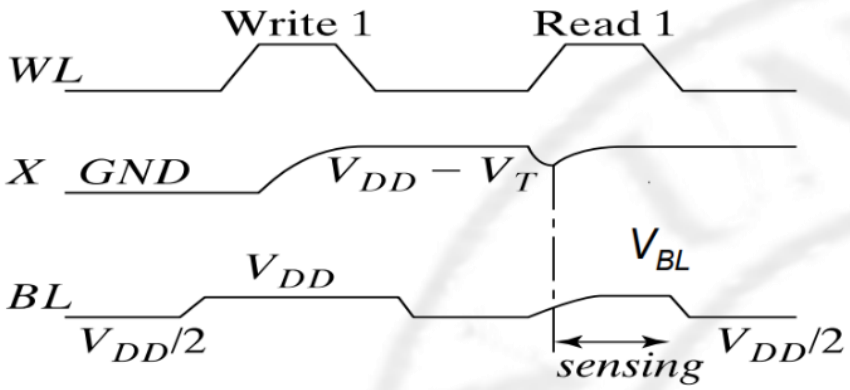
Trench Cell



Stacked-capacitor Cell

DRAM的读写操作





V_{BIT}= 0 or ($V_{DD} - V_T$)

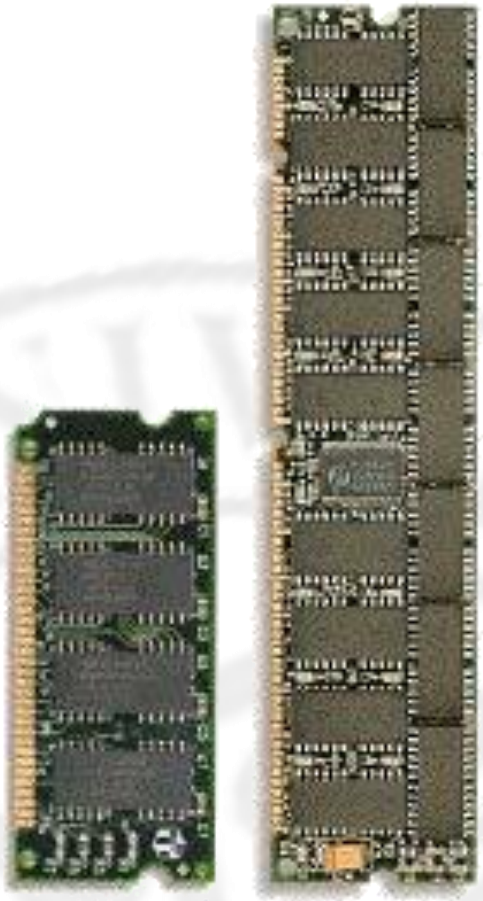
Write: C_S is charged or discharged by asserting WL and BL.
Read: Charge redistribution takes places between bit line and storage capacitance

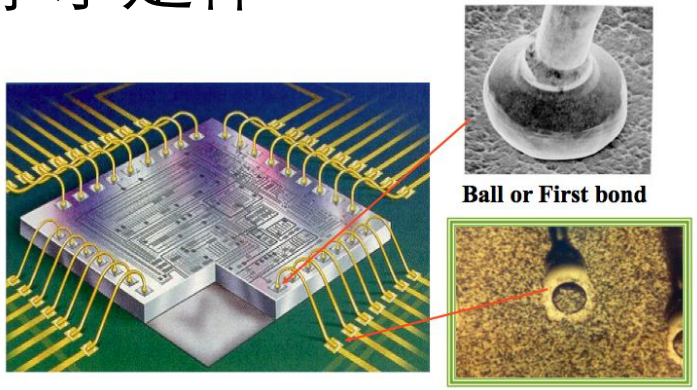
$C_S << C_{BL}$ Voltage swing is small; typically around 250 mV.

- To get sufficient C_s, **special IC process is used**
- Cell reading is destructive, therefore read operation always is followed by a write-back

DRAM封装 I – 2D

- DIMM(双列直插内存模块)包含具有并行时钟/控制/地址信号的多芯片(有时需要缓冲器将信号驱动到所有芯片)
- 数据引脚并排返回宽字节
  - e.g.,64-bit 数据线使用 16*4-bit
- 单芯片打线/压焊不遵循摩尔定律



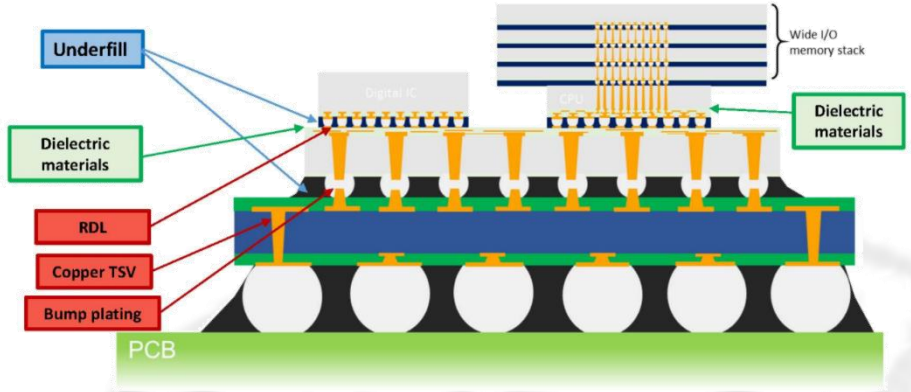


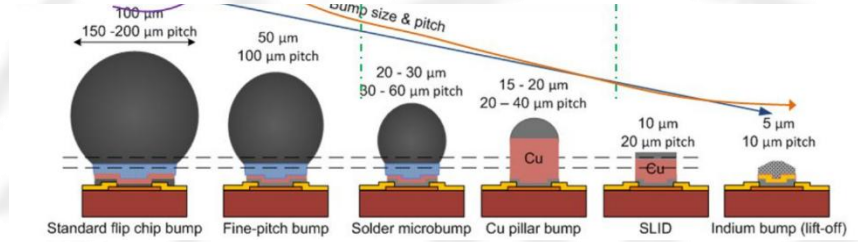
Ball or First bond

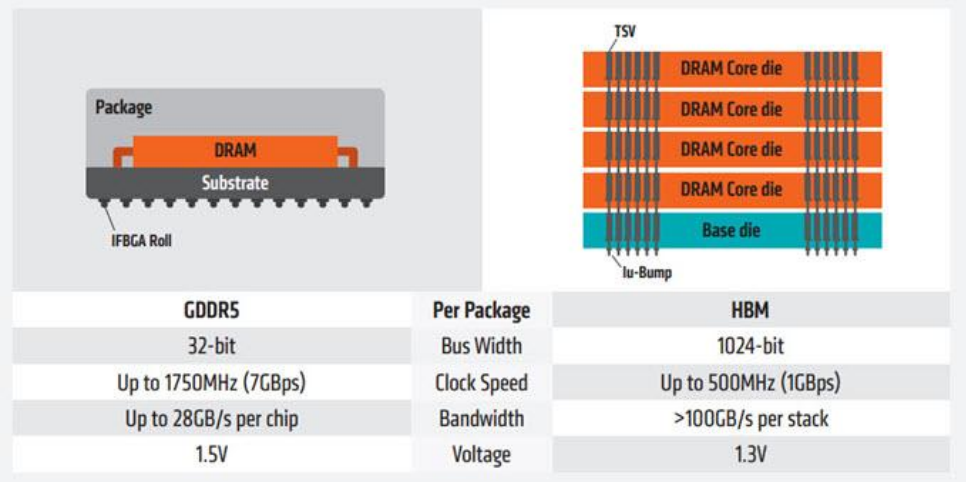
Stitch, Crescent or 2nd Bond

DRAM封装 II – 3D/2.5D

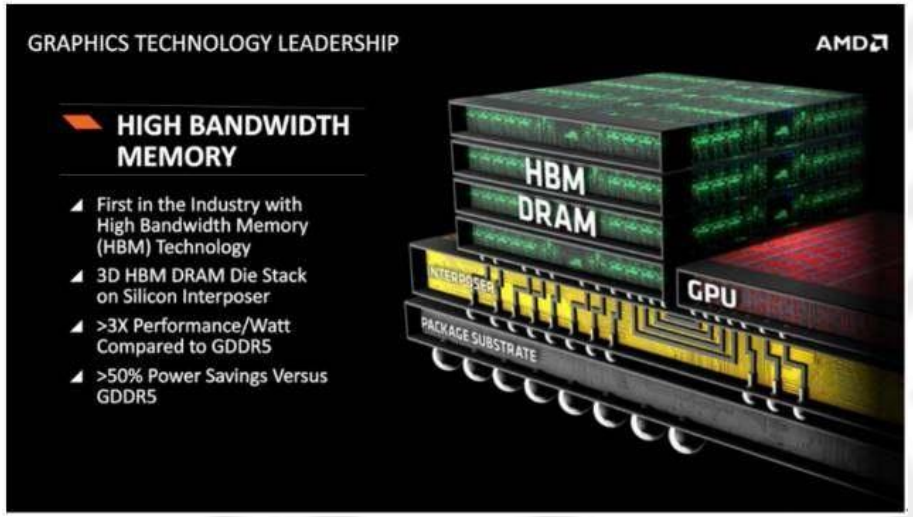
- 倒装片焊盘按摩尔定律缩放
- TSV(硅通孔)技术进一步提高了单位面积晶体管的密度







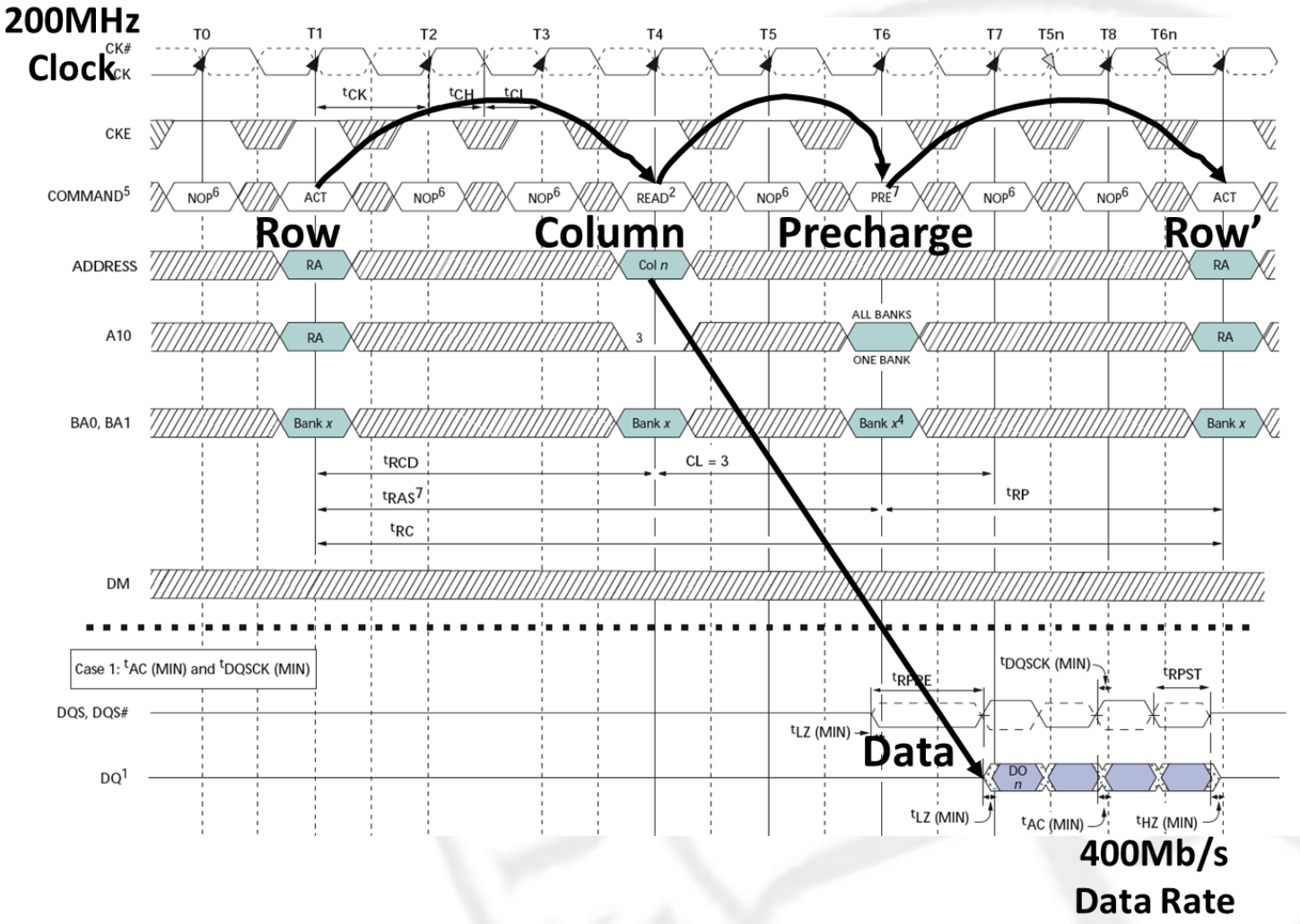
GDDR5	Per Package	HBM
32-bit	Bus Width	1024-bit
Up to 1750MHz (7GBps)	Clock Speed	Up to 500MHz (1GBps)
Up to 28GB/s per chip	Bandwidth	>100GB/s per stack
1.5V	Voltage	1.3V



DDR和处理器-存储瓶颈

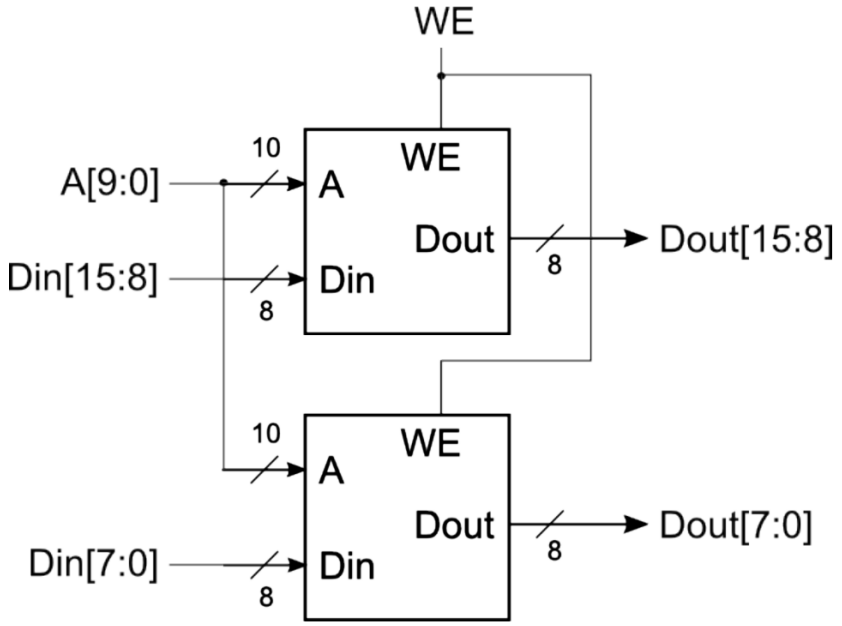
- 双倍数据速率 Double Data Rate(DDR)

- 高速计算机性能通常受存储带宽和延迟的限制
  - 延迟(单次访问时间)
  - 带宽(单位时间访问次数)

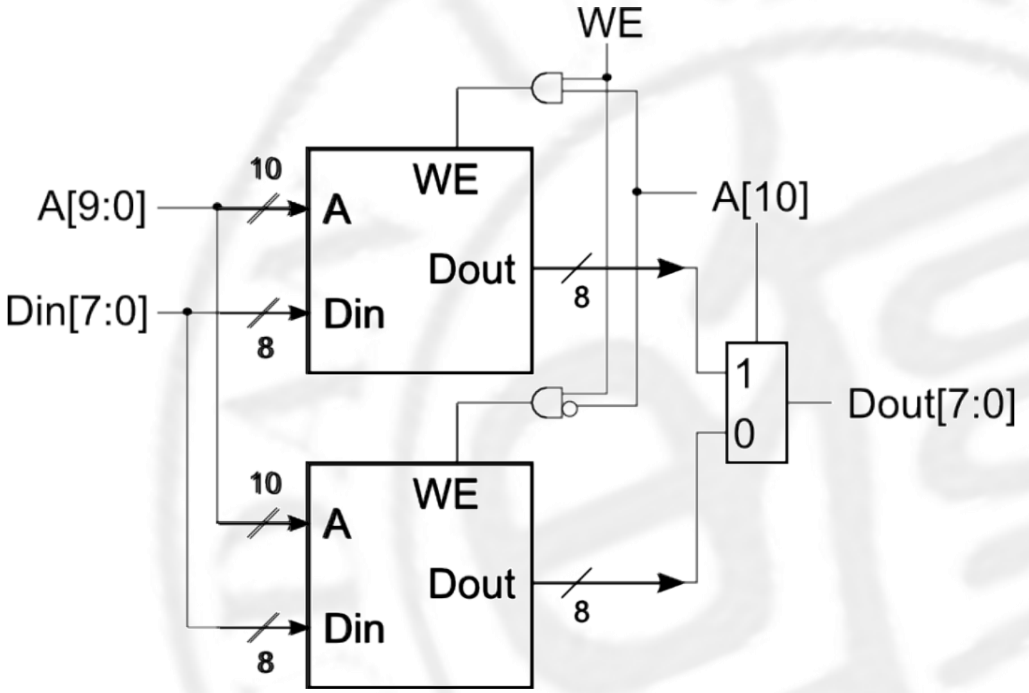


多个存储器的级联

- 拓展位宽

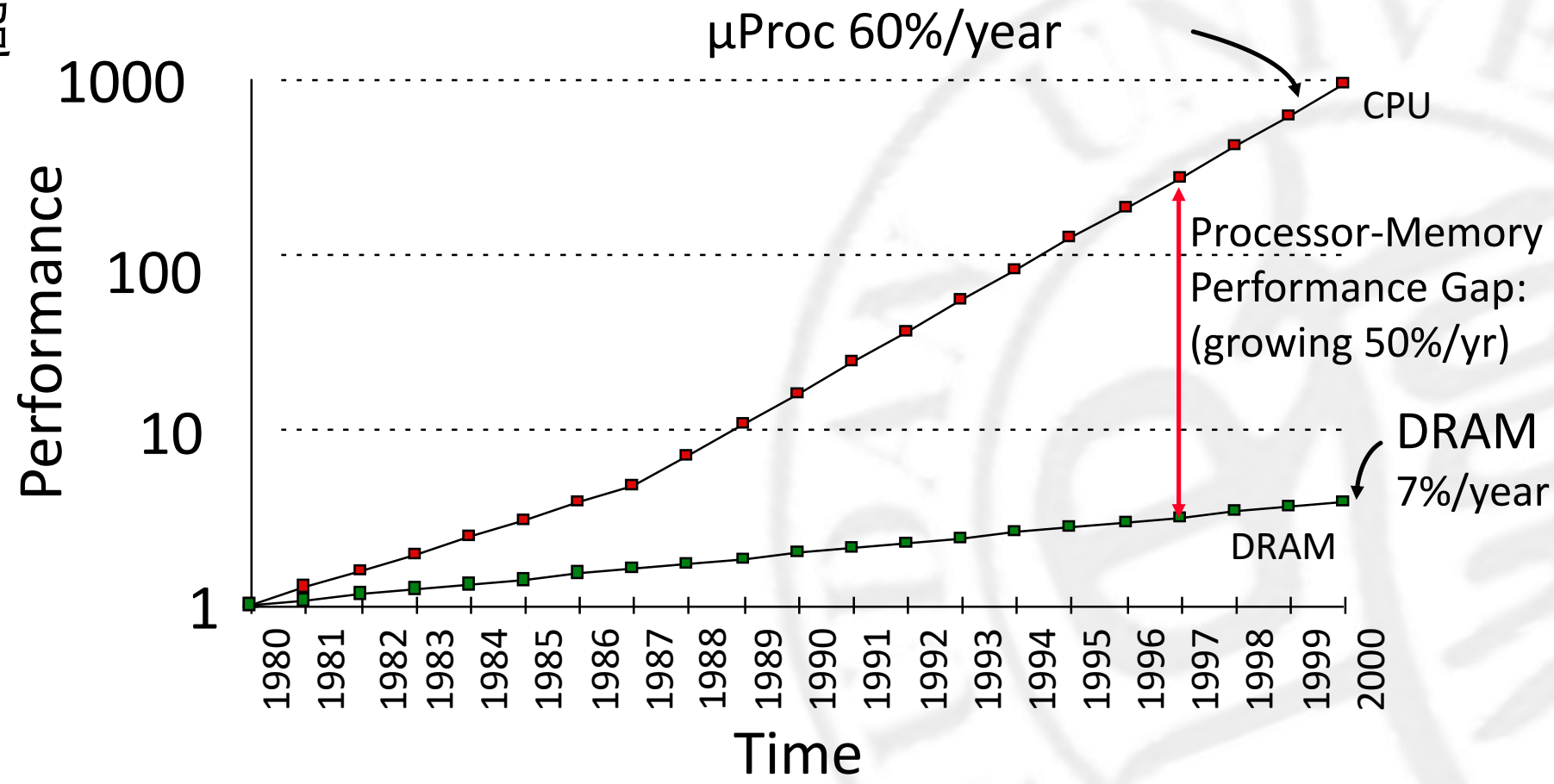


- 拓展深度

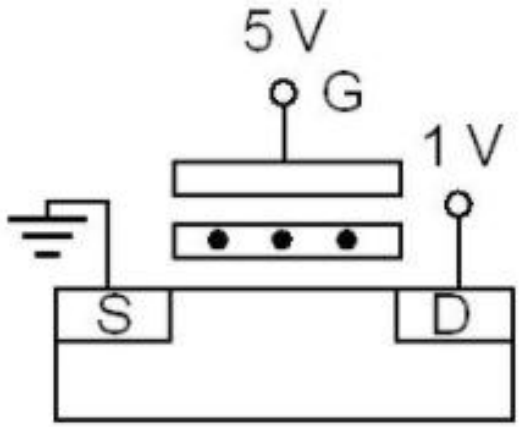


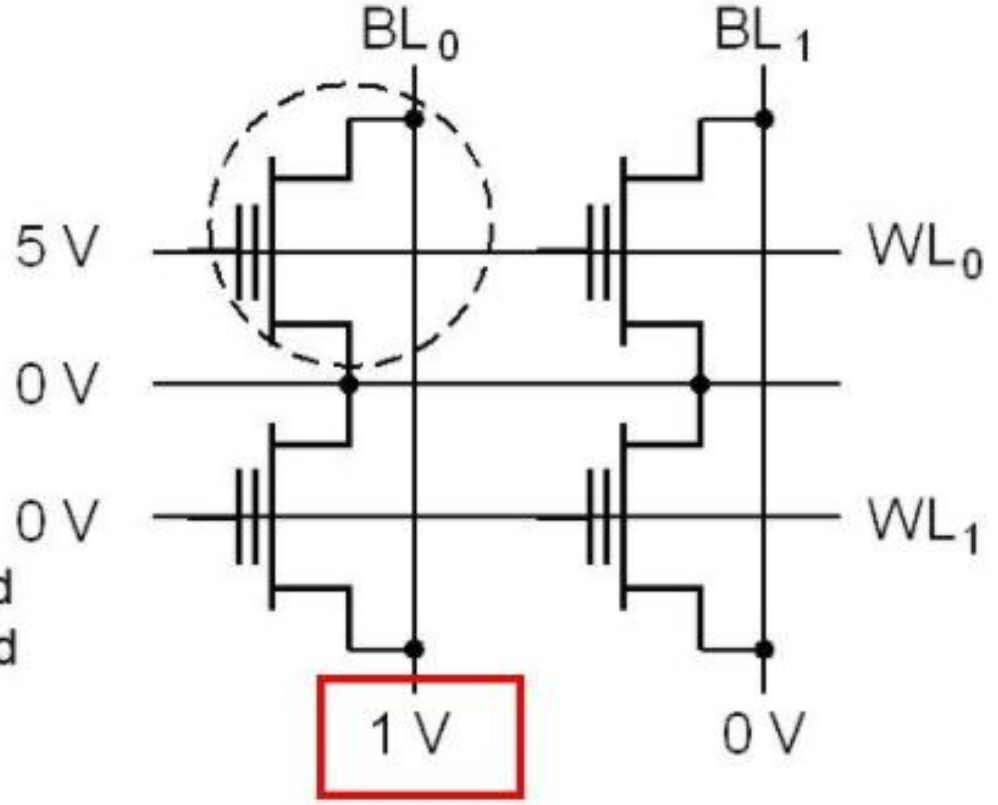
处理器-存储间的差距(延时)

- 3GHz 4-issue 超标量处理器
- 100ns DRAM
- 1200 指令



NOR Flash – 读操作



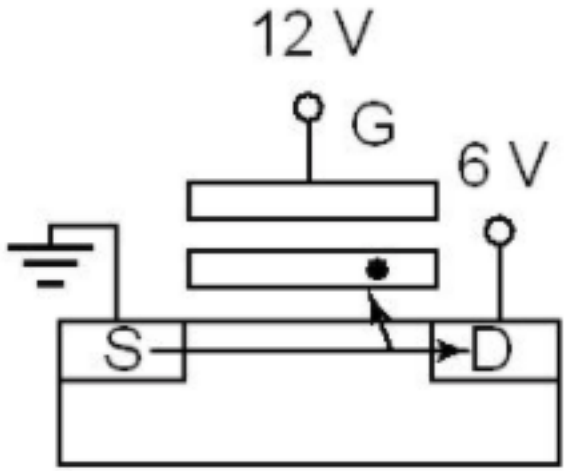


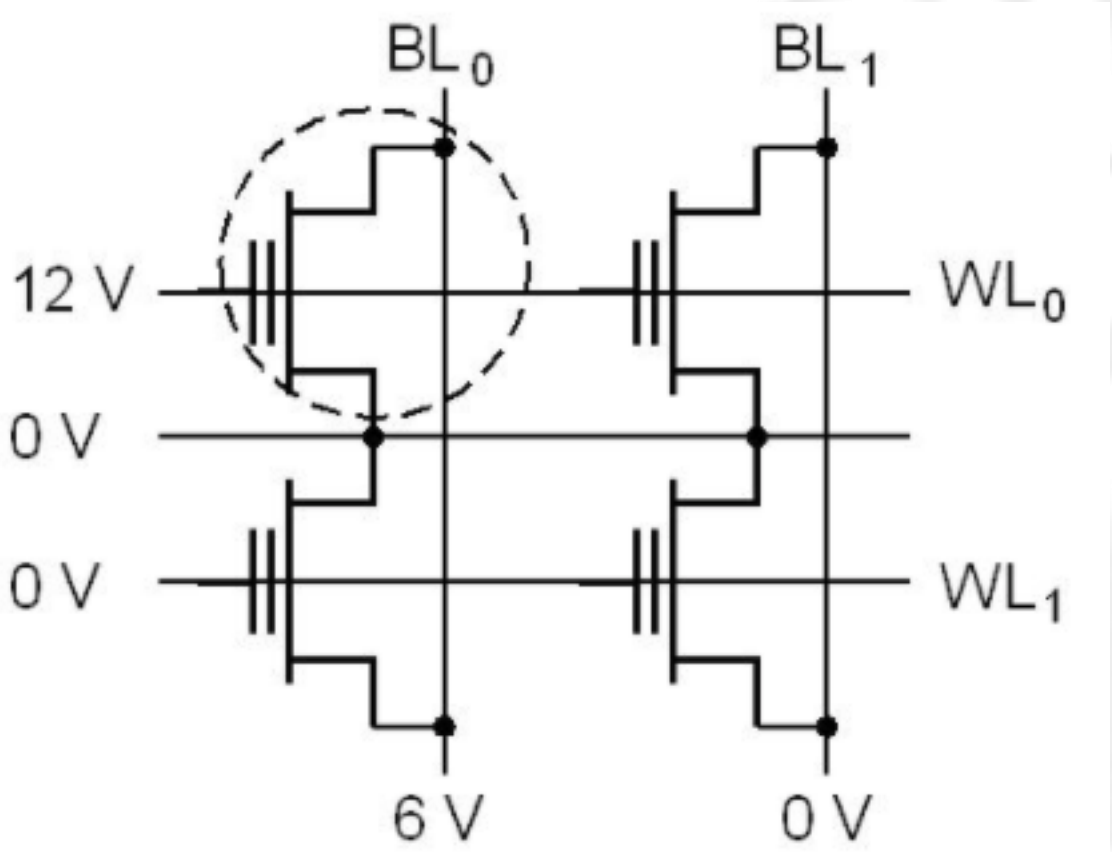
In read operation the programmed transistor stores 1 as it is switched off permanently

NOR Flash memories have

- **Fast random read time**
- **Slow erasure and programming time**
- **Need precise control of thresholds**

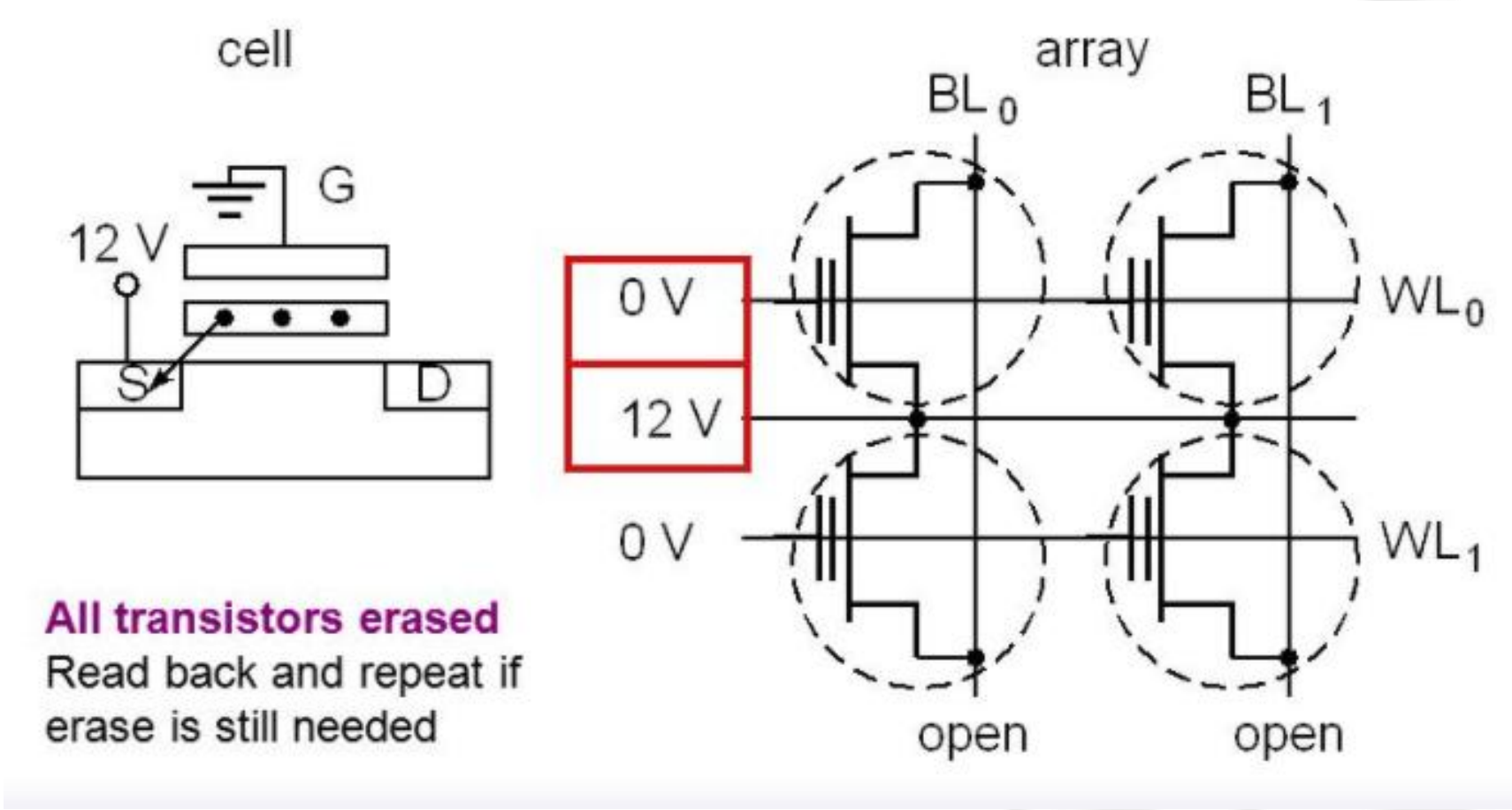
NOR Flash – 写操作





读写操作的电源电压不同

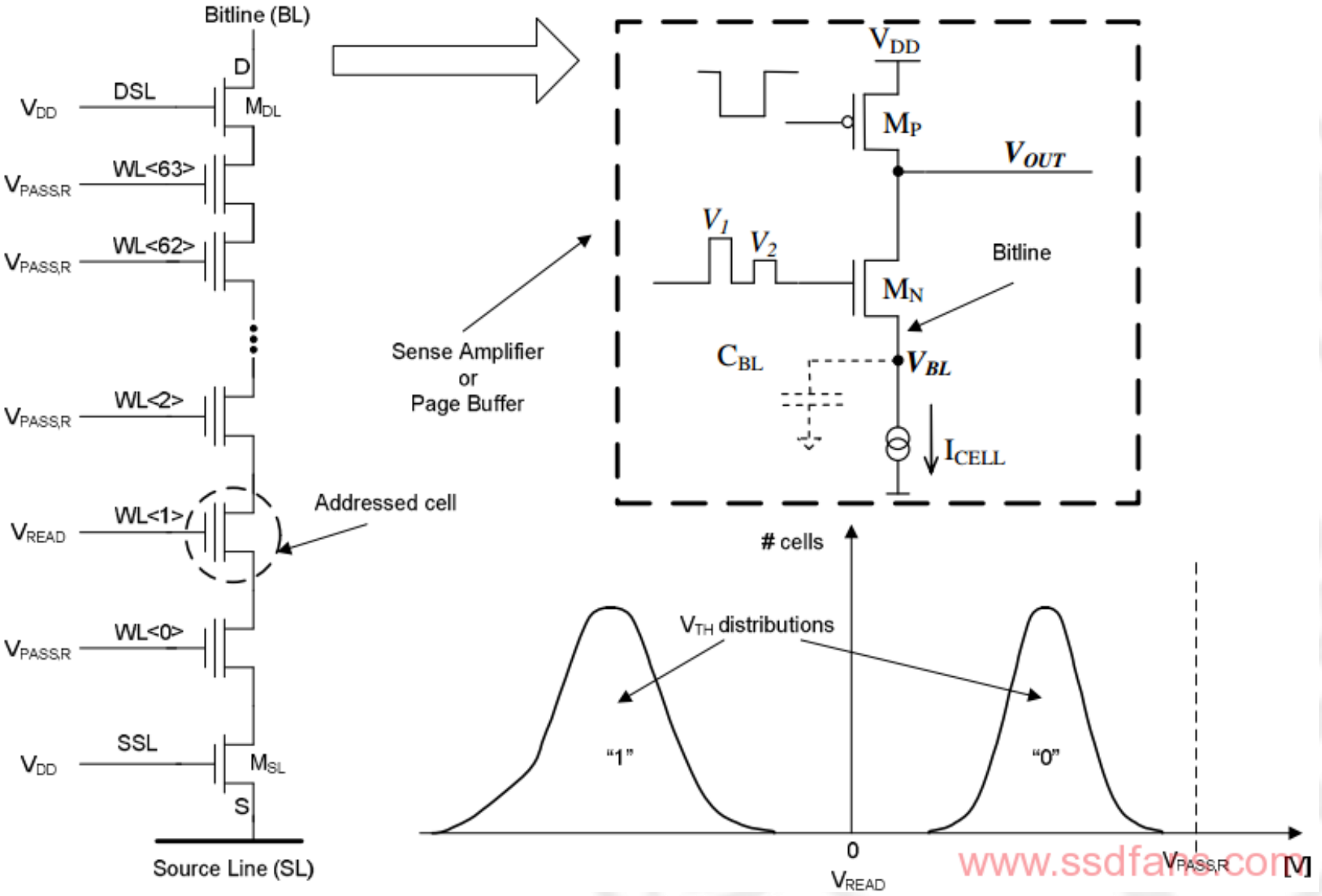
NOR Flash – 擦除操作



格式化——关注Source line

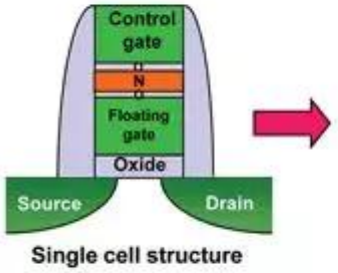
NAND Flash

- 当我们读一个cell时它的gate施加电压Vread(0V)，其他cells的gate端加Vpass,R电压(通常4-7V)以便无论cells的Vth值大小都可以保证其他cells完全导通开启。

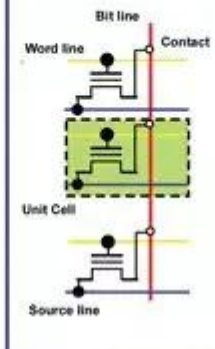
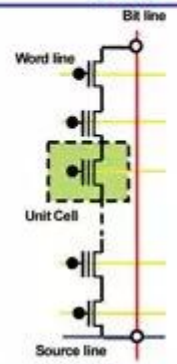
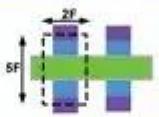
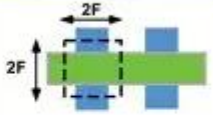


Flash SSD

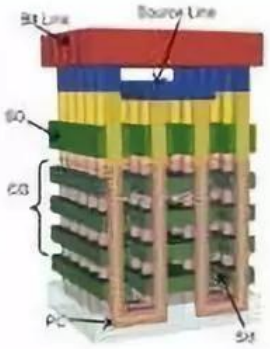
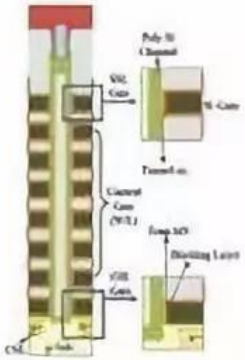
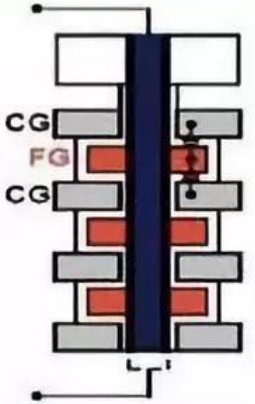
- NAND Vs. NOR



Single cell structure

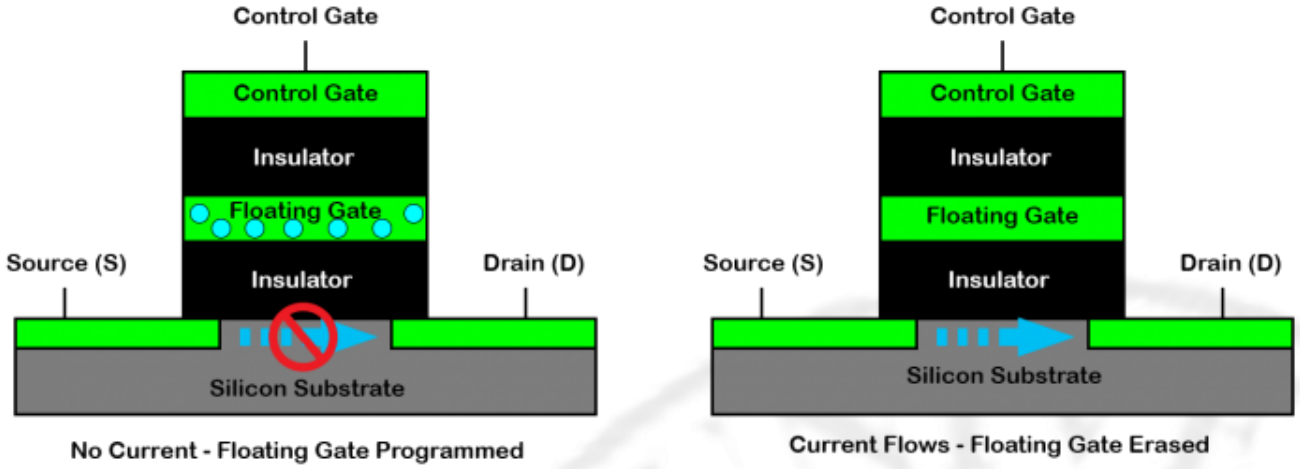
	NOR	NAND
Cell Array		
Layout		
Cross section	 $10F^2$	 $4F^2$

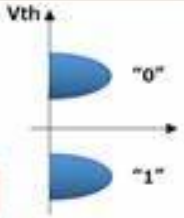
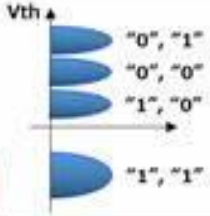
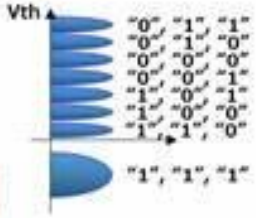
- 3D Flash

	p-BiCS (Toshiba)	TCAT (Samsung)	3D FG (Hynix)
Structure	 Tanaka. H, VLSIT 2007	 J. Jang, VLSIT 2009	 S. Whang, IEDM 2010
Key Features	- P+ SONOS Cell	- TANOS Cell	- Floating Gate
Key Issue	- Large Cell Size - Reliability	- Large Cell Size - SL Resistance	- Process of bit separation - Disturbance

Flash SSD

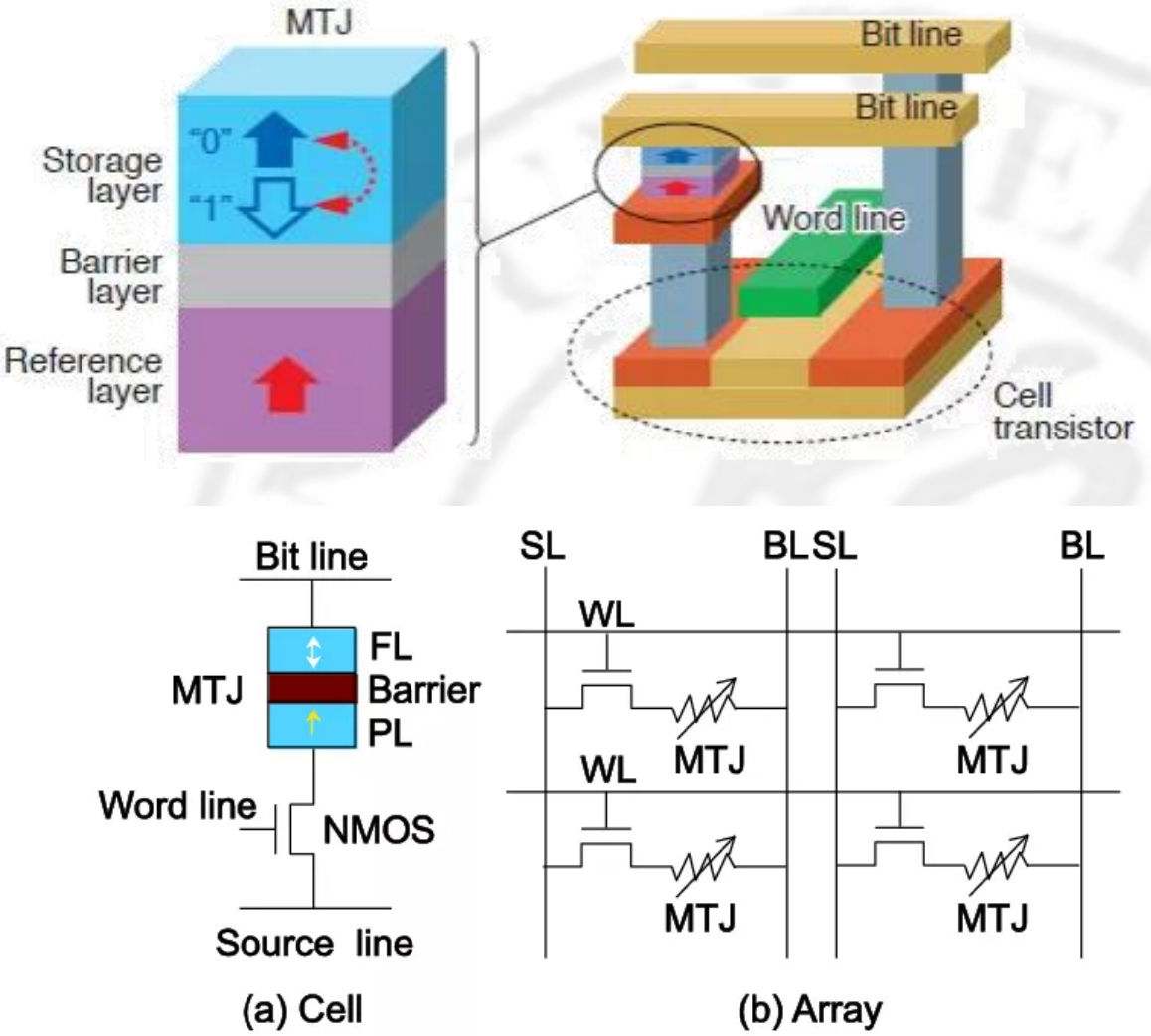
- 在浮栅中充电
- 单值单元 vs. 多值单元



Device Type	Stored Information / Memory Cell	State Count	Vth Distribution of the memory cell
SLC (Single Level Cell)	**1** bit / cell	**2**	
1 bit per cell → Reliable, Higher cost			
MLC (Multi Level Cell)	**2** bits / cell	**4**	
2 bits share same cell → doubled Capacity , less reliable			
TLC (Triple Level Cell)	**3** bits / cell	**8**	
3 bits share same cell → Higher Capacity , poor reliable			

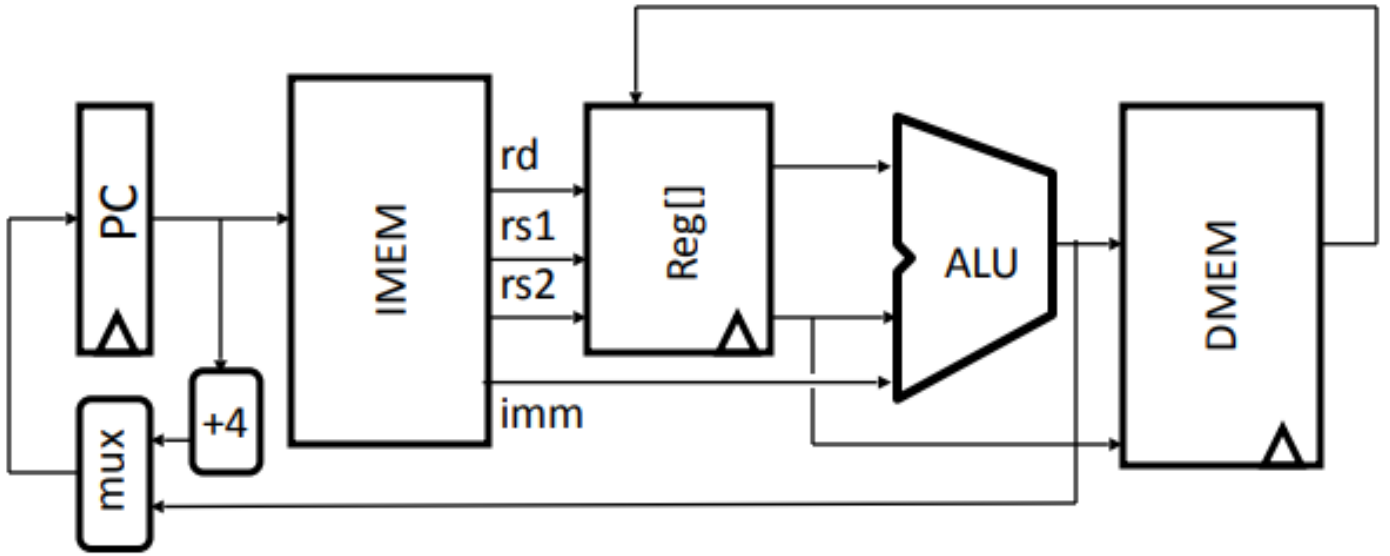
新原理存储器 – MRAM/ReRAM

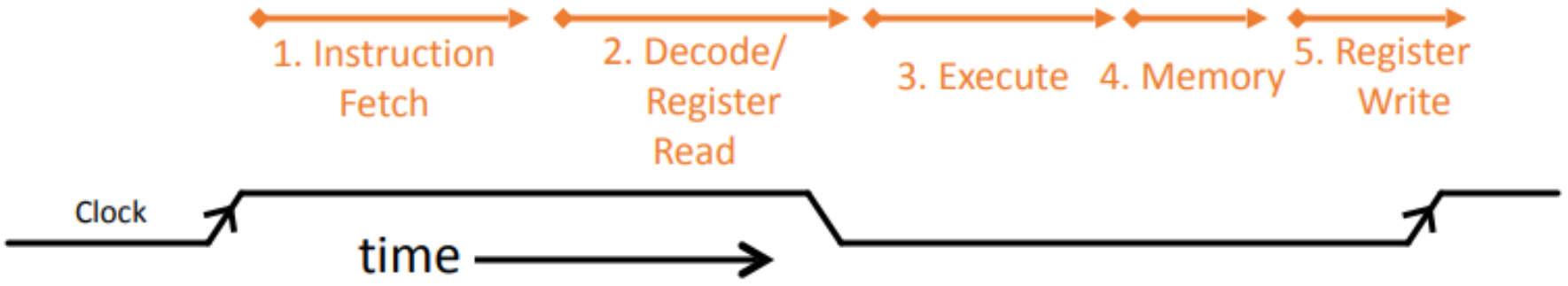
- 易失与非易失存储
- 与DRAM一样快/密集，与SRAM一样兼容CMOS，与闪存一样非易失?
- STT-MRAM(Spin Transfer Torque Magnetic RAM)自旋转移扭矩随机存取存储器
- 使用忆阻器的ReRAM



基于数字电路的处理器设计

处理器是用于按照规定执行顺序计算机指令集的数字电路





数字电路遇到的全新挑战

General purpose processors (CPUs) can **NOT** afford the recent high performance computing brought by fast growing AI algorithms.