

模拟与数字电路

Analog and Digital Circuits



课程主页 扫一扫

第 十六讲： **寄存器与存储器**

Lecture 16: **Register Files and Memories**

主 讲： 陈 迟 晓

Instructor : Chixiao Chen

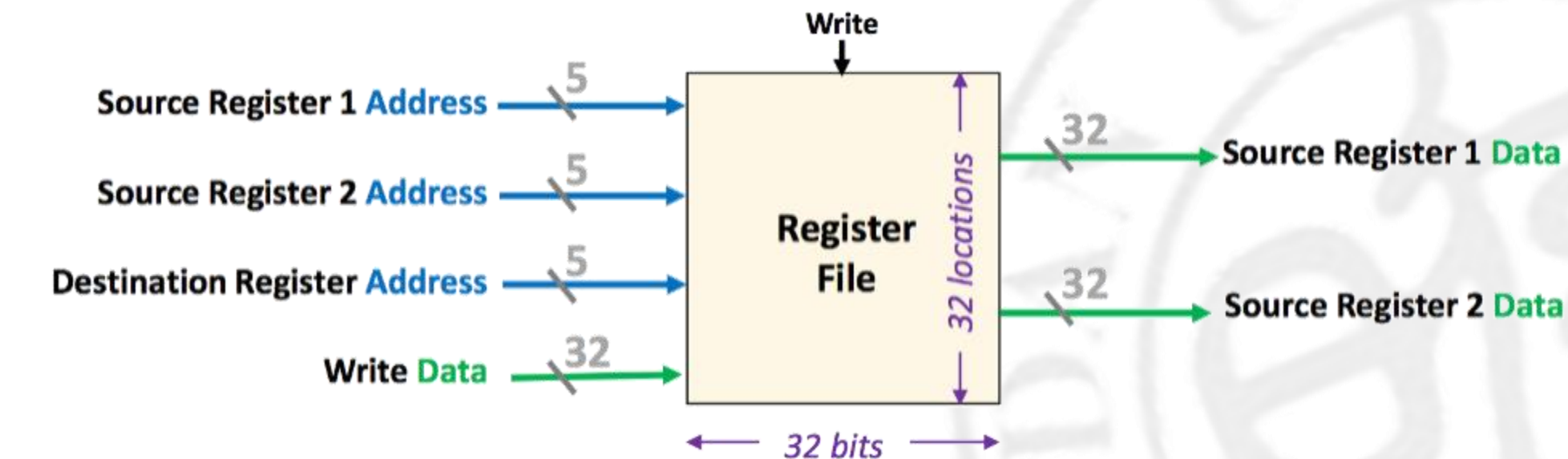
提纲

- 复习
 - 状态机的模型有那些，有什么区别？
- 寄存器、寄存器组、寄存器列表
- 存储器电路与结构



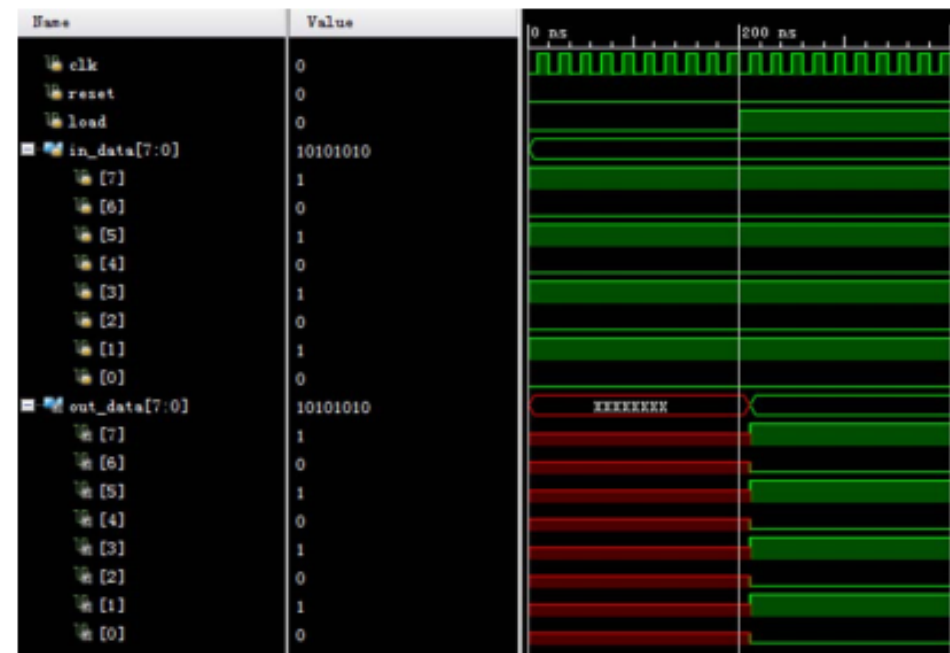
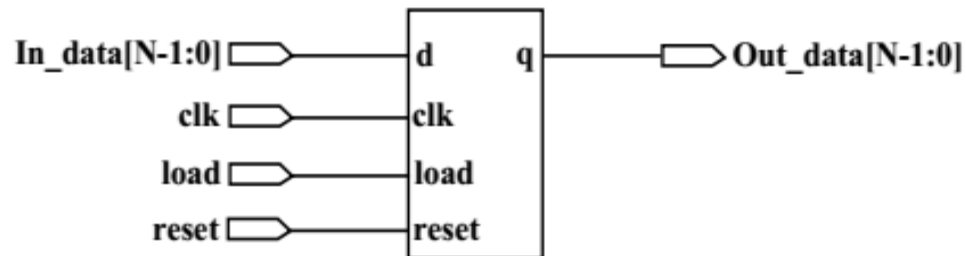
寄存器列表 Register File

- 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。

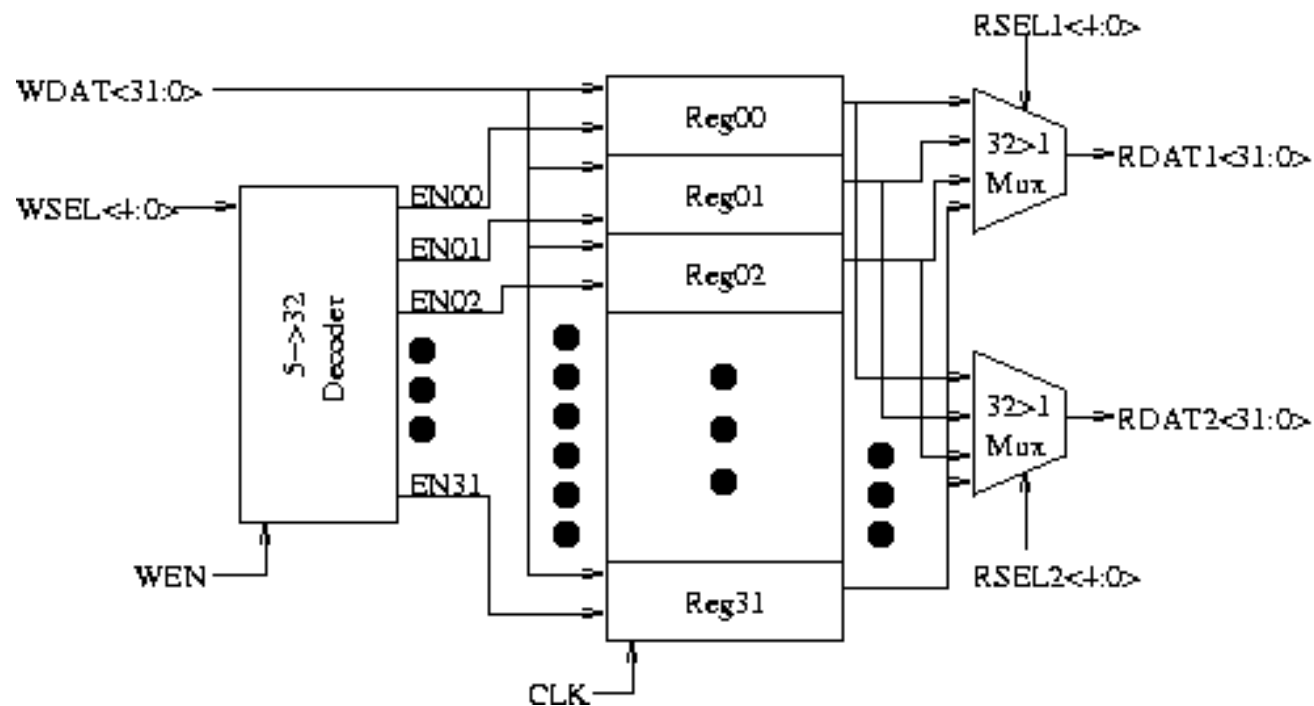
```
module reg_file
#(
parameter N = 8, //比特数
W = 2) //地址比特数
(
input clk,
input wr_en,
input [W-1:0] w_addr,r_addr,
input [BN-1:0] w_data,
output [BN-1:0] r_data
);
reg [BN-1:0] array_reg[2**W-1:0];
```

```
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

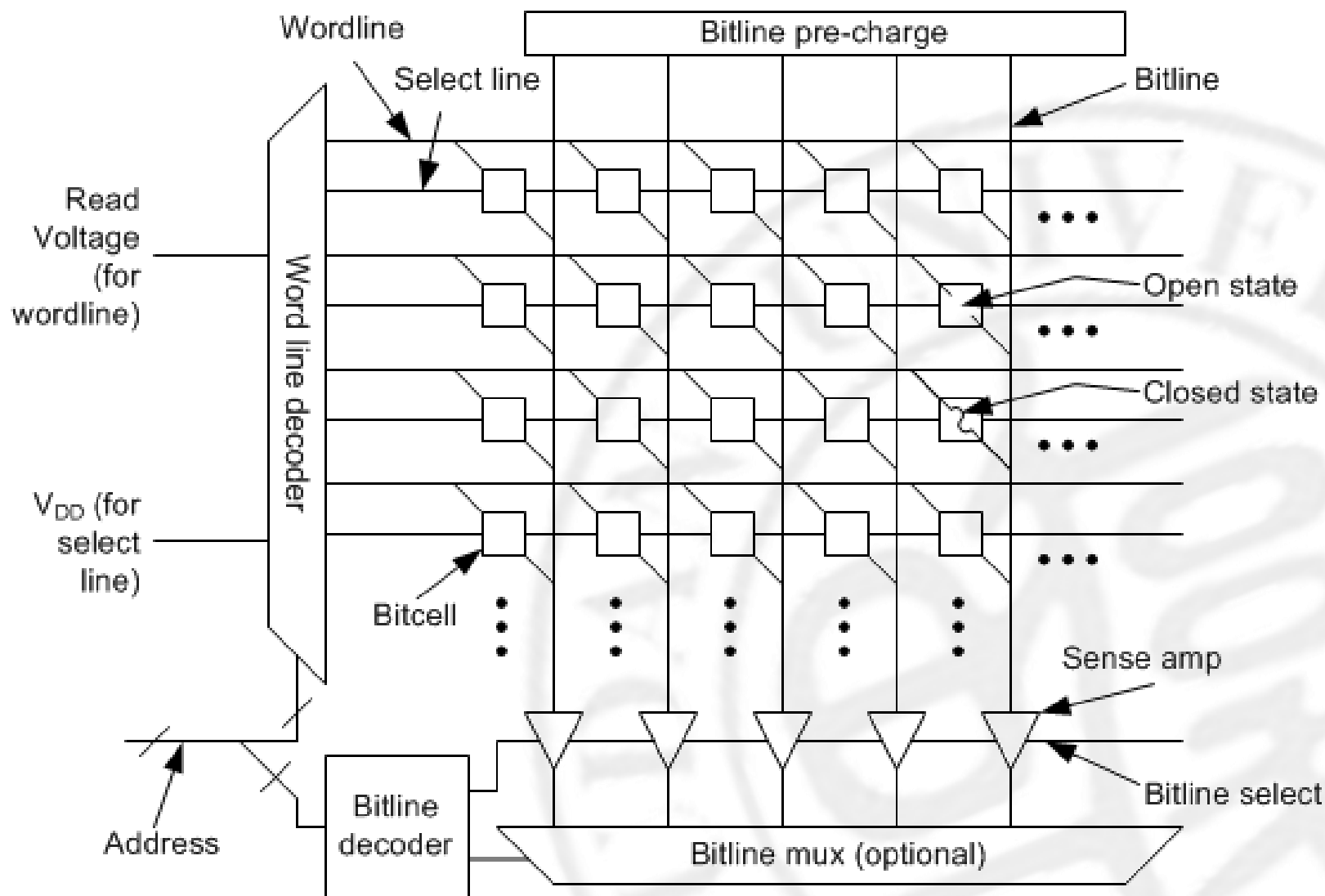
    always @(posedge clk)
        if (we3) rf[wa3] <= wd3; //write

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

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

存储器

- 通过二维阵列实现
- Word line (位线)
- Bit line (字线)
- Select line



地址译码器

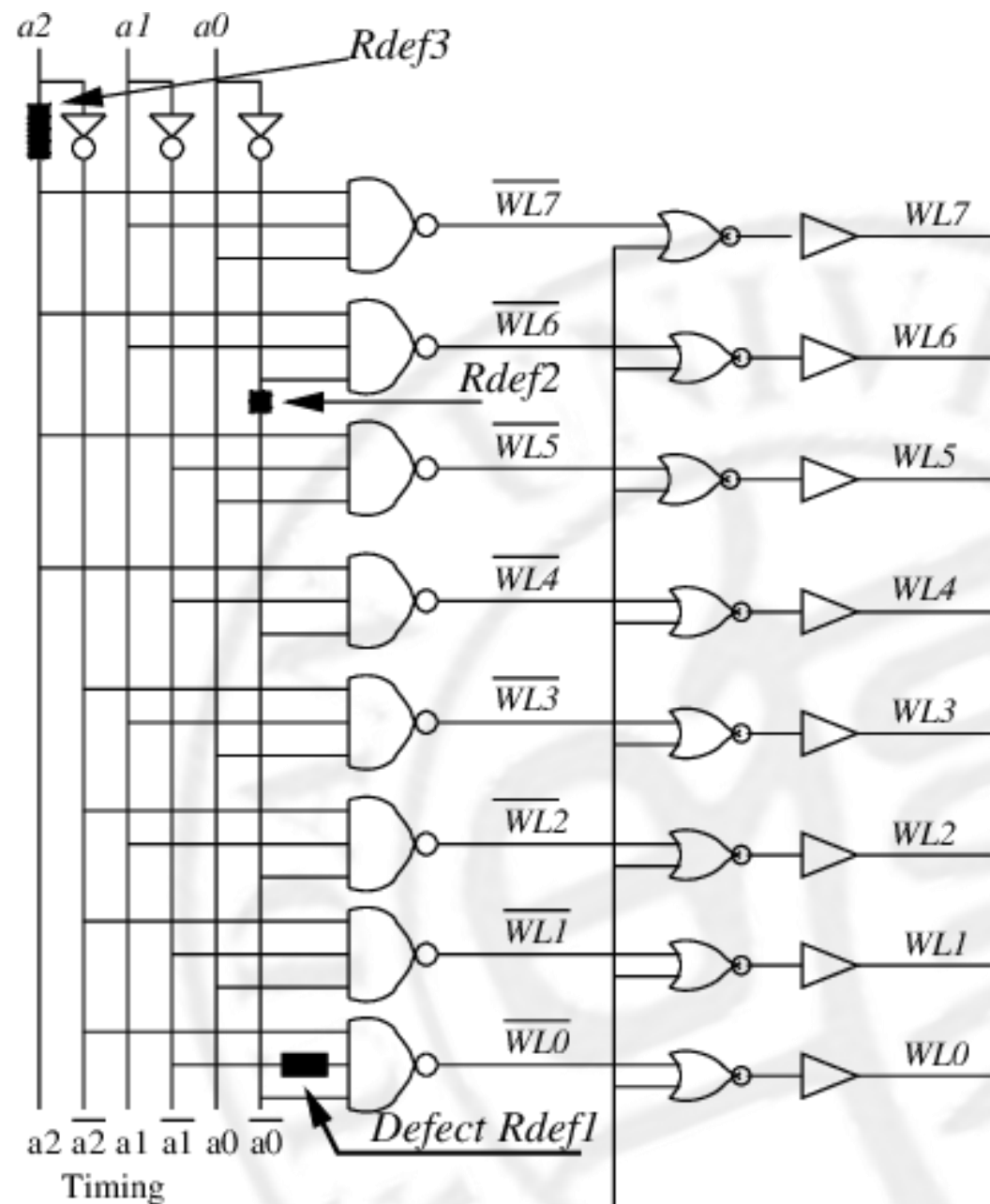
• 3-8译码器

```

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



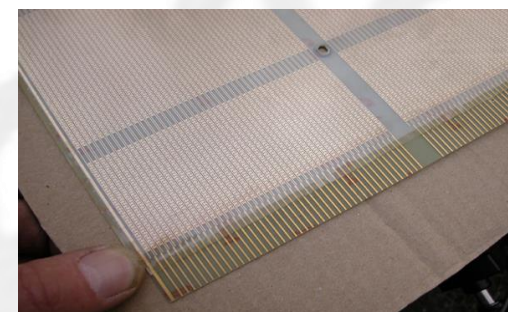
Punched cards, From early 1700s through Jacquard Loom, Babbage, and then IBM



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

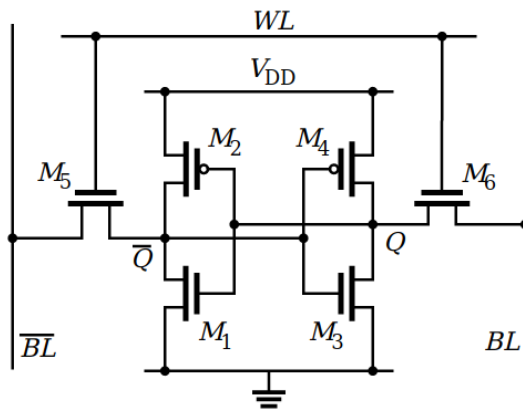


IBM Balanced Capacitor ROS

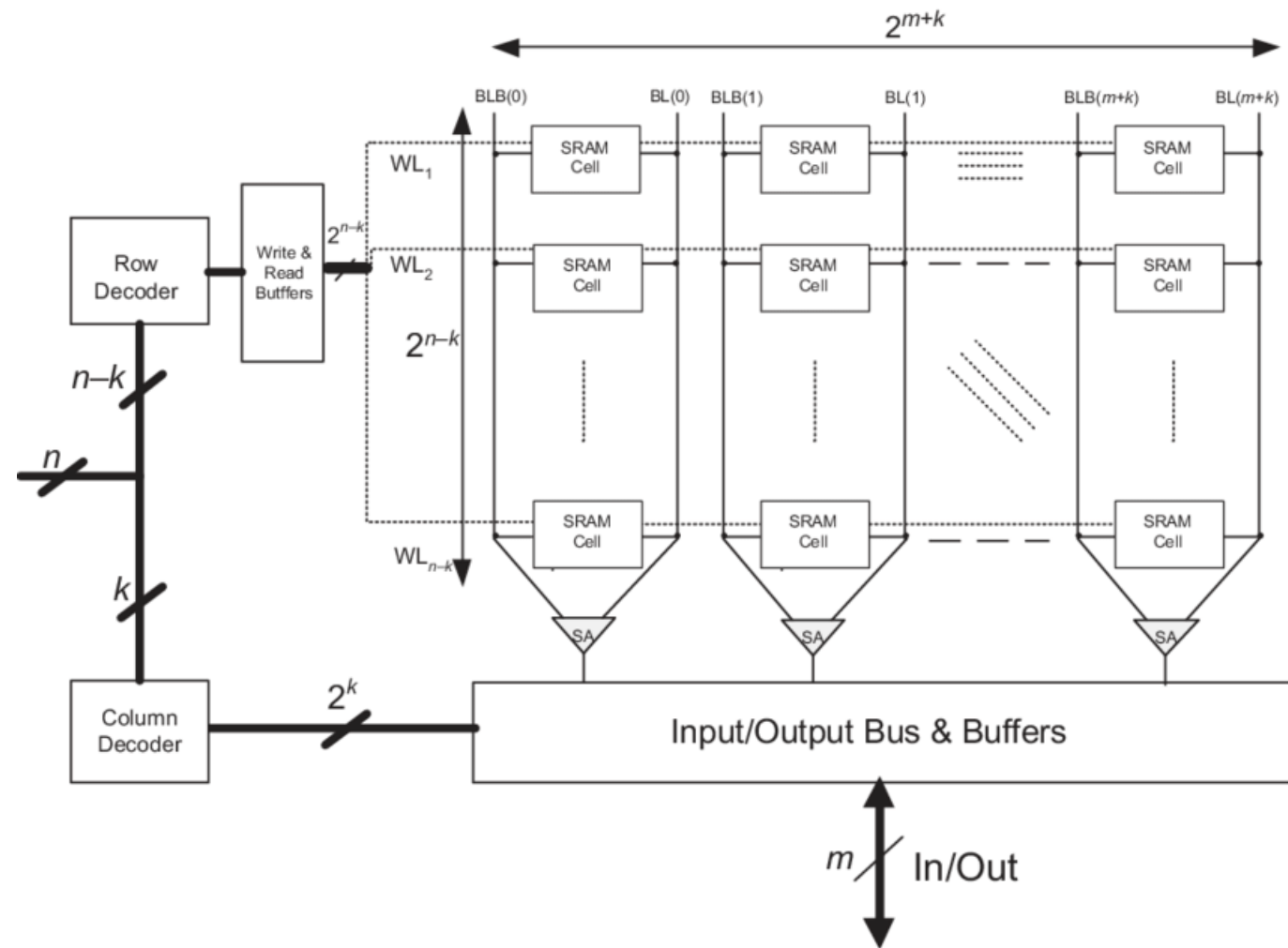
片上存储

- 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

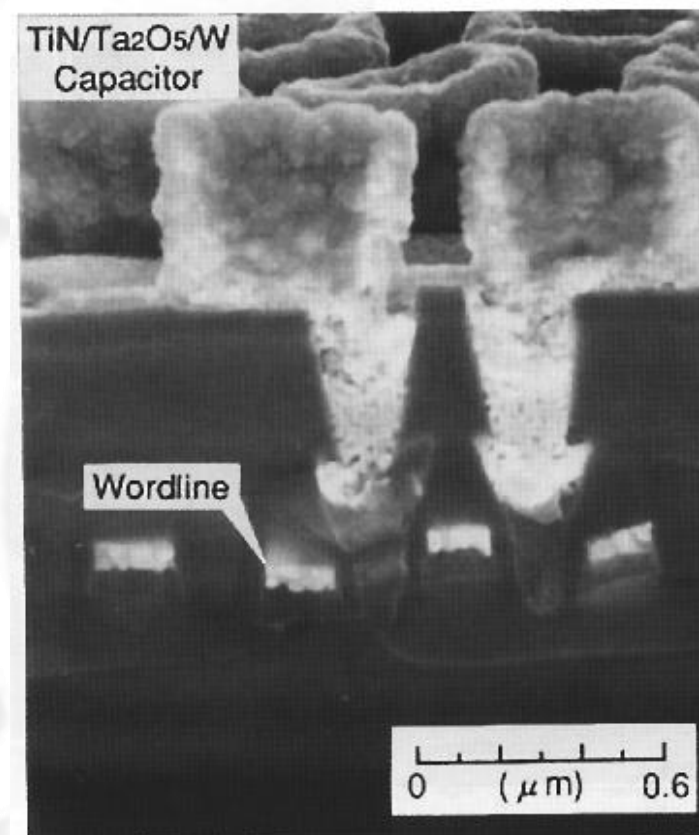
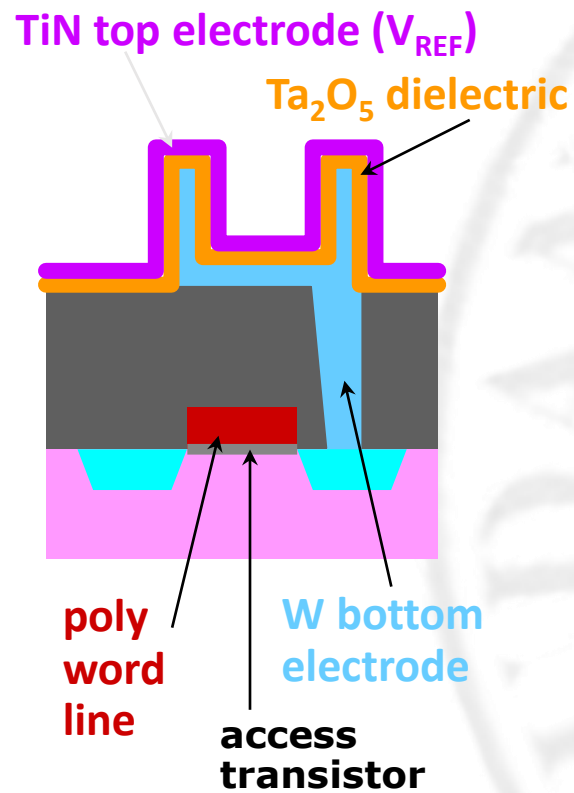
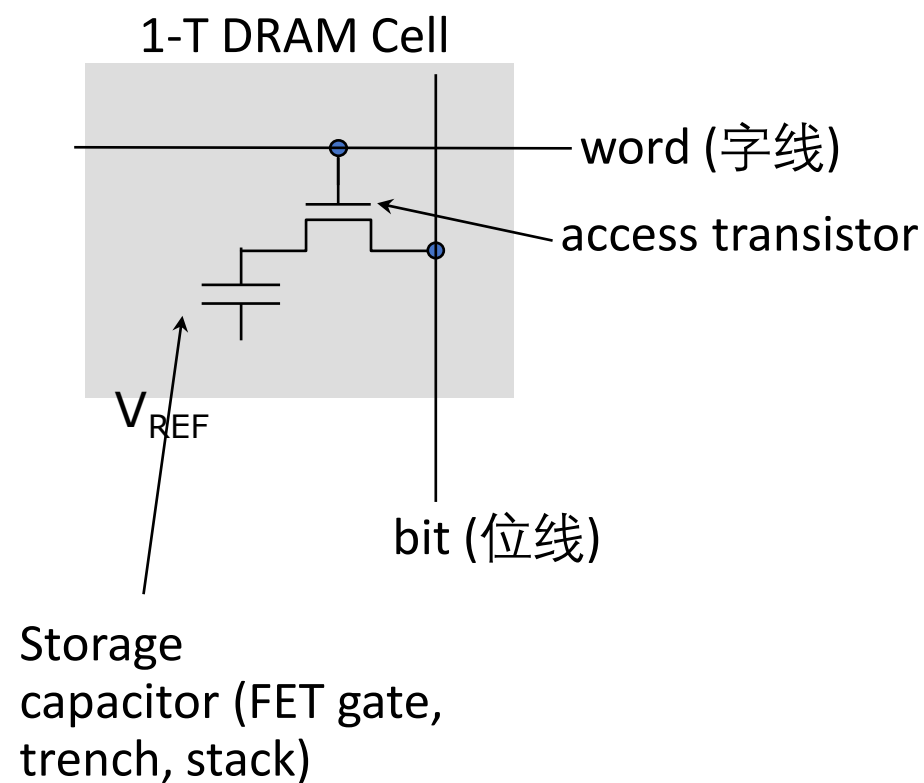


SRAM Memory Cell

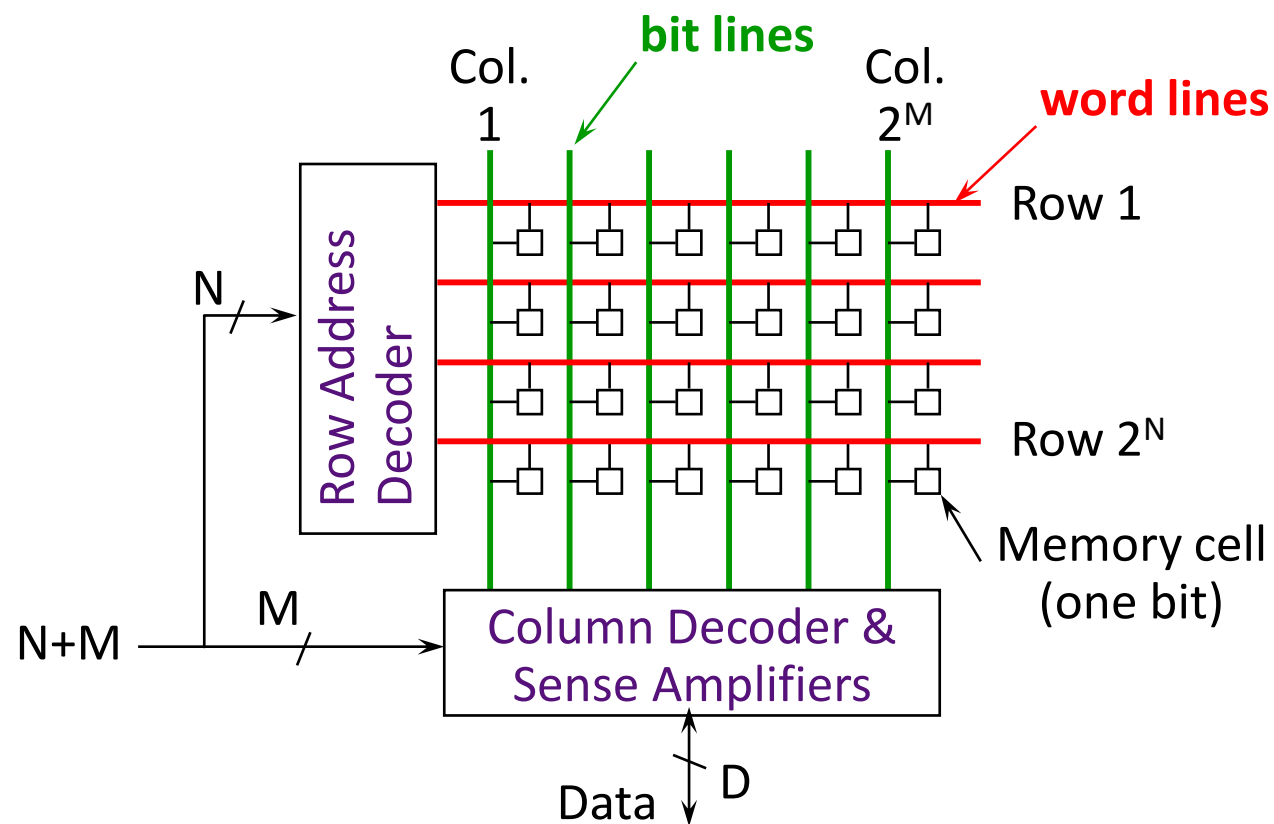


单晶体管动态RAM [Dennard, IBM]

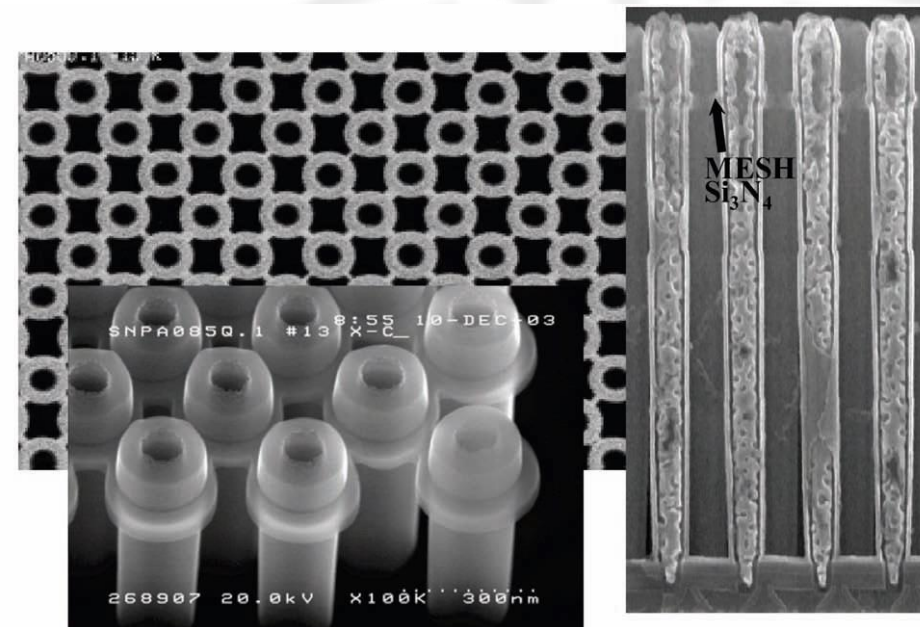
- 用一个电容存储 0/1， 用一个开关选择



现代DRAM架构/结构



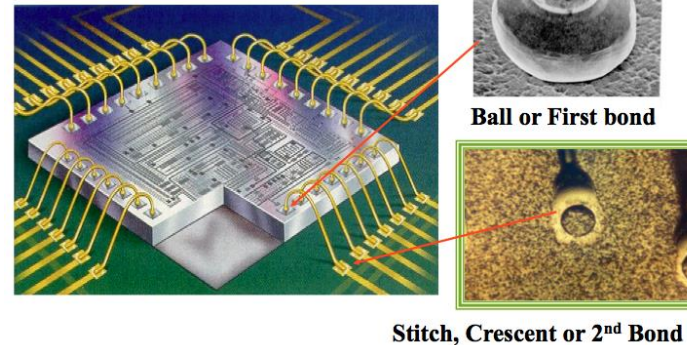
- 比特存储在二维片上阵列中



- 问题： DRAM vs. CMOS?

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



Stitch, Crescent or 2nd Bond



72-pin SO DIMM

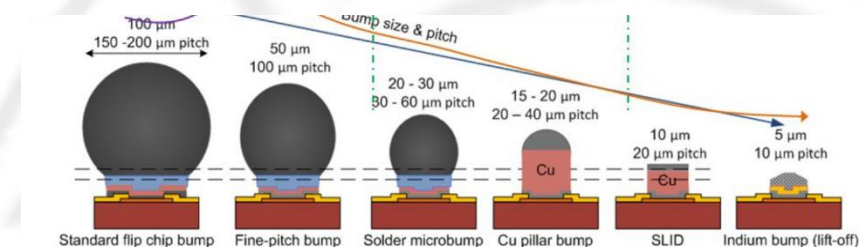
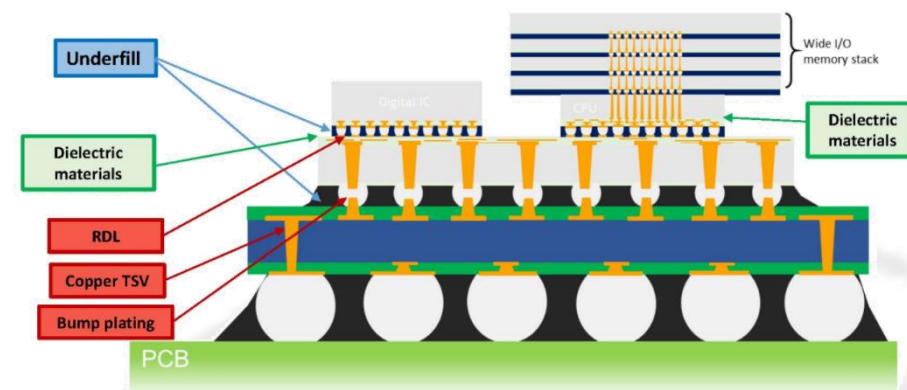


168-pin DIMM



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		HBM	
Package	IFBGA Roll	TSV	1u-Bump
32-bit		1024-bit	
Up to 1750MHz (7GBps)		Up to 500MHz (1GBps)	
Up to 28GB/s per chip		>100GB/s per stack	
1.5V		1.3V	

GRAPHICS TECHNOLOGY LEADERSHIP

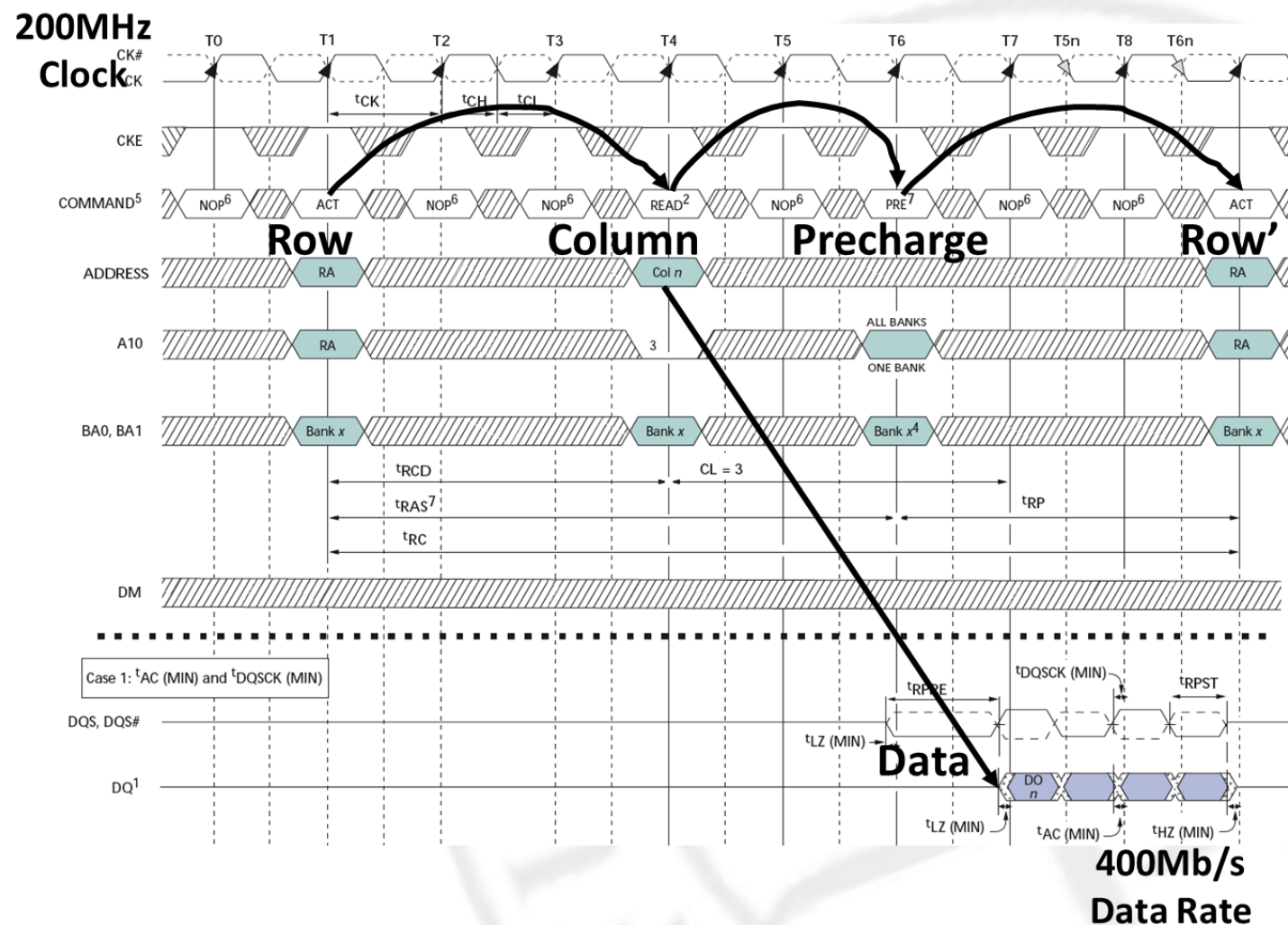
HIGH BANDWIDTH MEMORY

- First in the Industry with High Bandwidth Memory (HBM) Technology
- 3D HBM DRAM Die Stack on Silicon Interposer
- >3X Performance/Watt Compared to GDDR5
- >50% Power Savings Versus GDDR5

The image shows a 3D rendering of an AMD GPU with a stack of HBM DRAM (High Bandwidth Memory) on a silicon interposer. The GPU is connected to a CPU. The diagram highlights the high bandwidth and power efficiency of HBM technology compared to GDDR5.

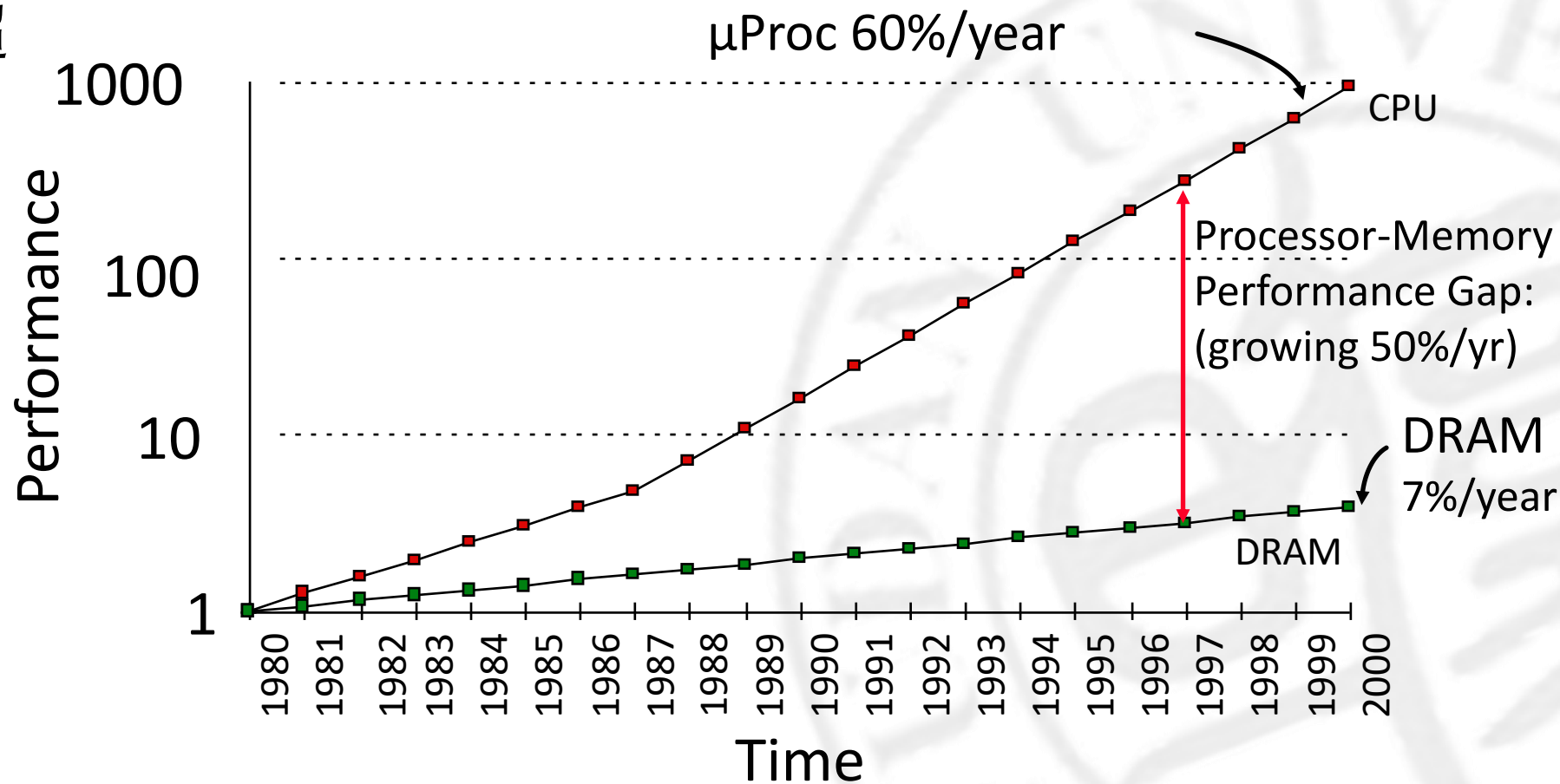
DDR和处理器-存储瓶颈

- 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)



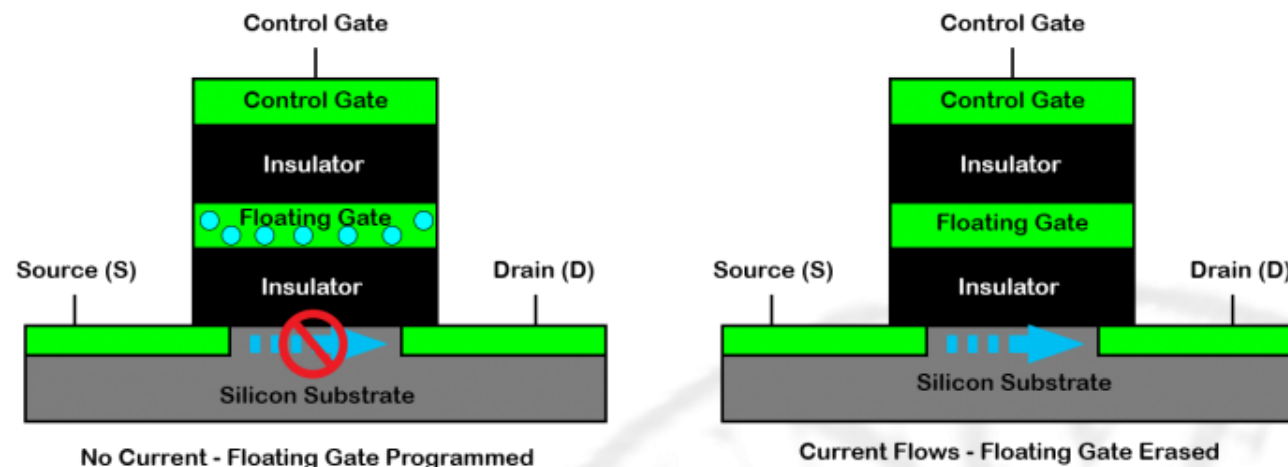
处理器-存储间的差距(延时)

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



Flash SSD

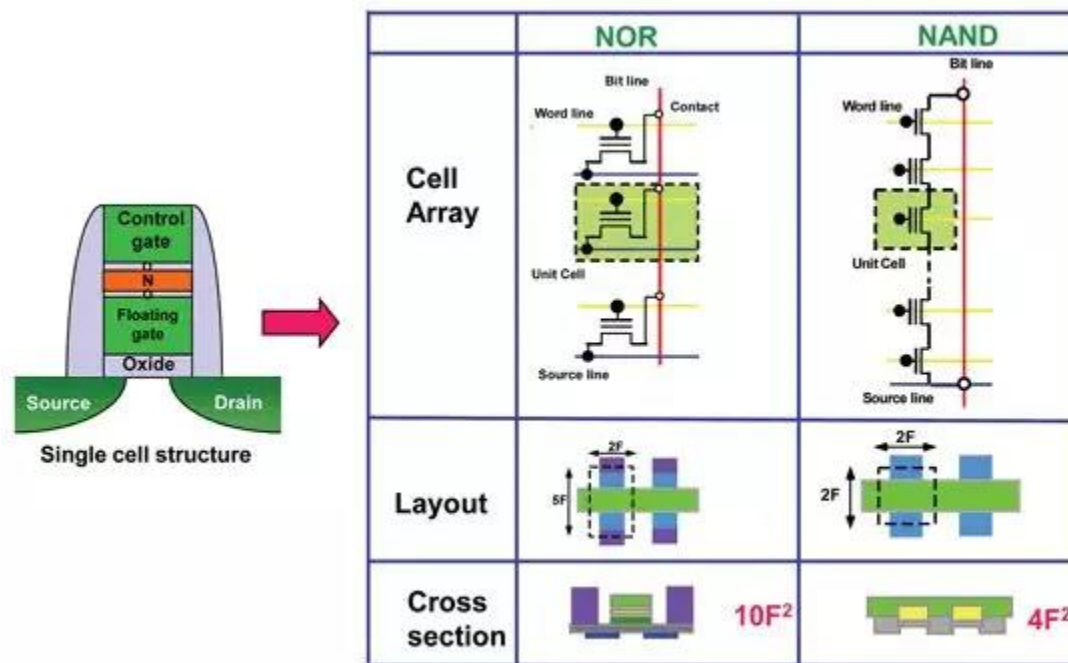
- 在浮栅中充电
- 单值单元 vs. 多值单元



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			

Flash SSD

- NAND Vs. NOR

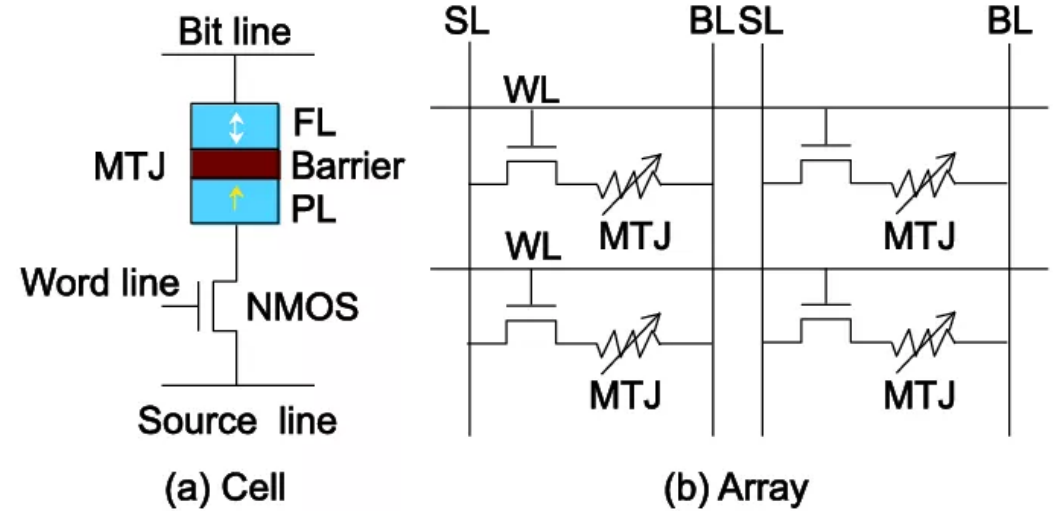
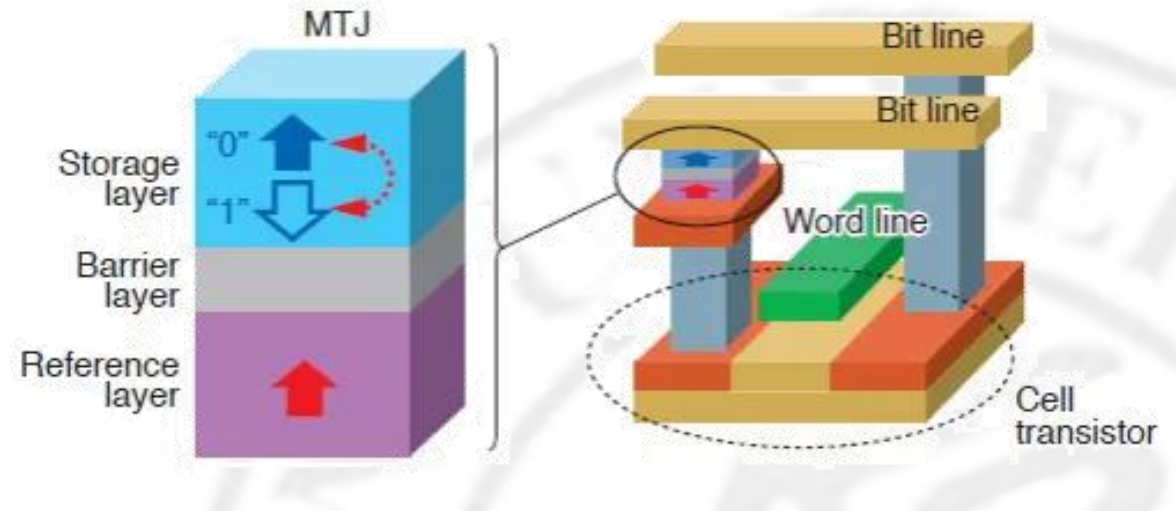


- 3D Flash

	p-BICS (Toshiba)	TCAT (Samsung)	3D FG (Hynix)
Structure	<p>Tanaka, H, VLSIT 2007</p>	<p>J. Jang, VLSIT 2009</p>	<p>S. Whang, IEDM 2010</p>
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

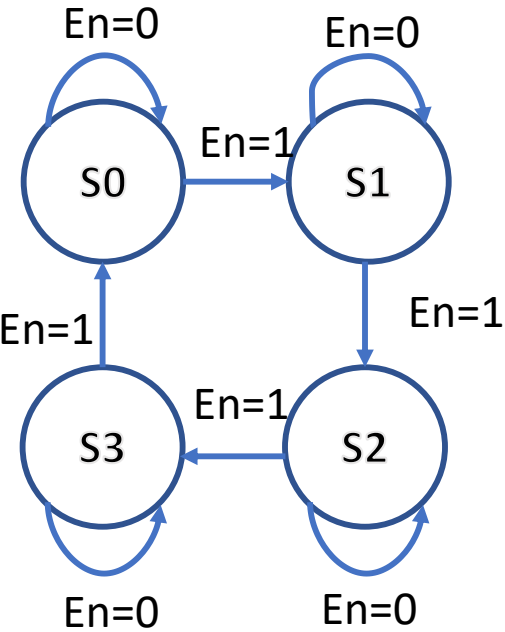
新原理存储器 – MRAM/ReRAM

- 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



作业与实验准备：

- 1.设计实验一： 设计一个简单的分频电路完成10分频；
- 2.设计实验二： 设计一个单一进制的计数器， 如十进制；
- 3.设计实验三： 设计一个一输入的Mealy型状态机， 该状态机有四个状态， 通过输入信号En的使能实现四个状态的循环切换。每个状态需输出不同的output,如下表。



State	Output[3:0]
S0	2
S1	4
S2	8
S3	10

要求： 上述三个实验须在周五课前完成行为级仿真； 如有任何问题可及时联系助教焦博解决。。