**Module 3: Memory**

Unit 3.1: Sequential Logic

1. Use the same hardware over time

Inputs change and outputs should follow

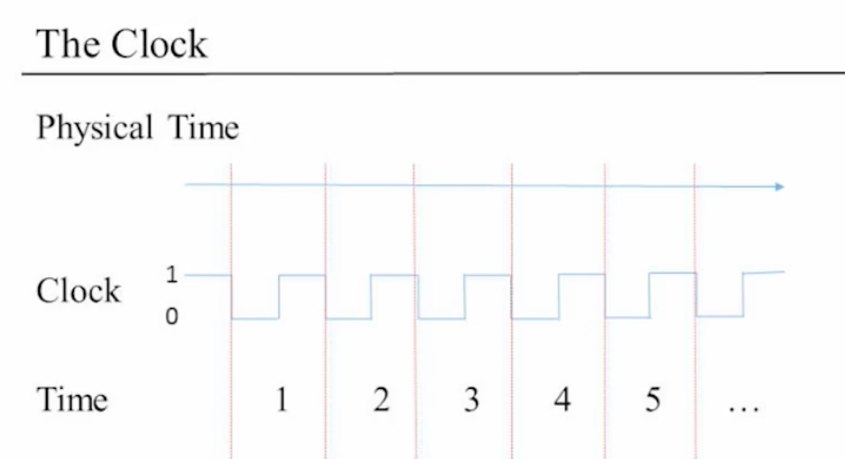
e.g.: for i = 1 to 100, a[i] = b[i] + c[i]

1. Remember “State”

Memory, Counters

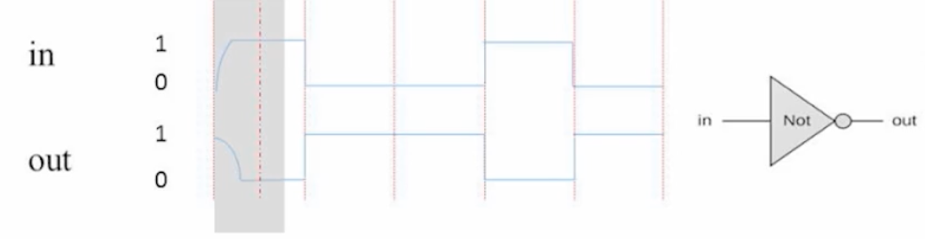
e.g.: for i = 1 to 100, sum = sum + i

1. The Clock: split physical time into discrete units



1. 之所以将时间划分为离散的单元，是因为不想让考虑time delay

Delay：只要灰色区域没有超过一个unit，那么灰色区域内可以忽略



1. Combinatorial Logic: out[t] = function(in[t])

Sequential Logic: out[t] = function(in[t-1])

remember things from the last period

图表

中度可信度描述已自动生成日历

描述已自动生成

recall it: state[t] = function(state[t-1]) // Iteration

Unit 3.2: Flip Flops

1. Missing ingredient:

remember one bit of information from time t-1 so it can be used at time t

1. At the “end of time” t-1, such an ingredient can be at either of two states:

“remembering 0” or “remembering 1”

1. This ingredient remembers by “flipping” between these possible states
2. Gates that can flip between two states are called Flip-Flops
3. The “Clocked Data Flip Flop”: out[t] = in[t-1]图示

   描述已自动生成

这里的三角形代表

一个sequential chip

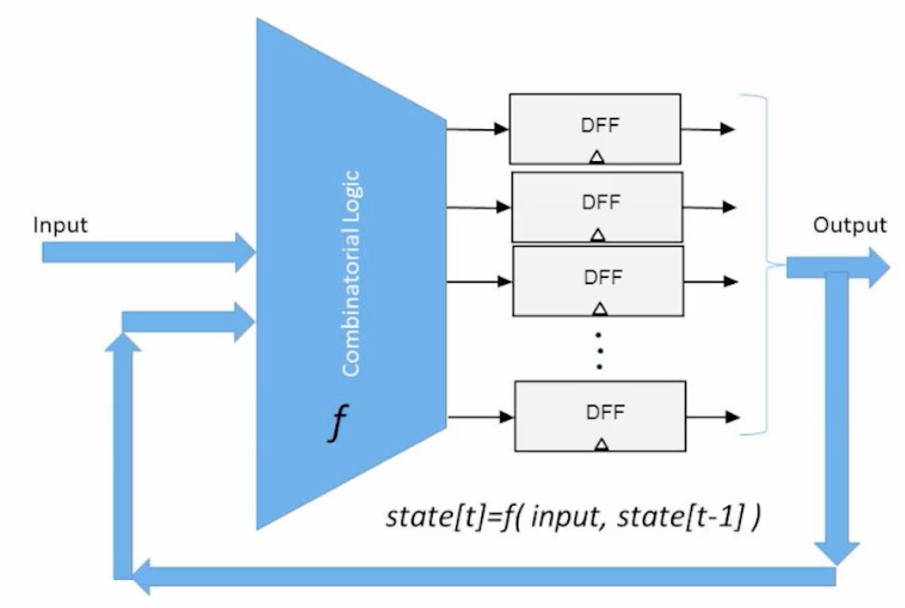
a chip that depend on time

1. In this course, the D Flip Flop is primitive
2. in many physical implementations, it may be built from actual Nand gates:

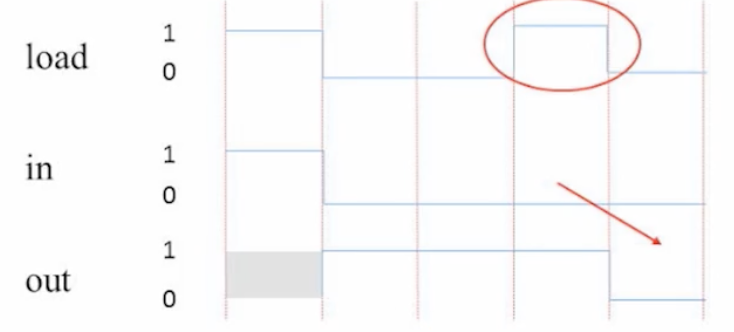
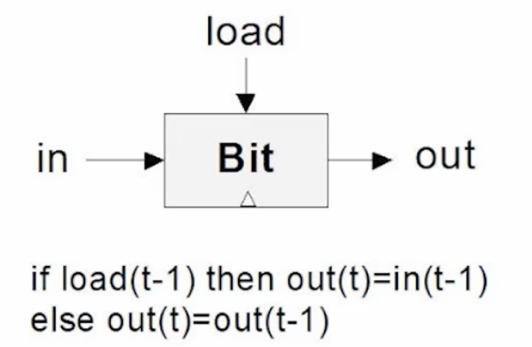
Step 1: create a “loop” achieving an “un-clocked” flip-flop

Step 2: Isolation across time steps using a “master-slave” setup

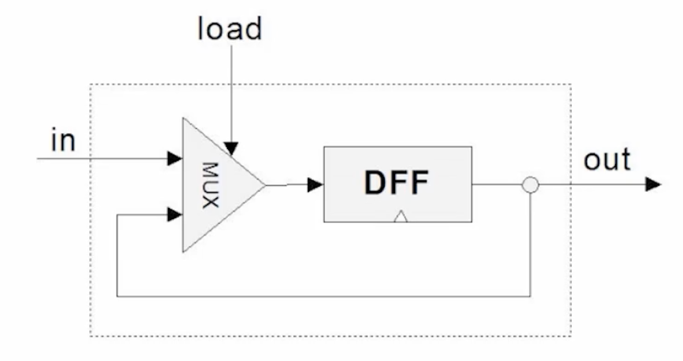
1. Our Hardware Simulator forbids “combinatorial loops”



1. Goal: remember an input bit “forever”, until requested to load a new value



1. 1-Bit Register, The most basic memory element



Unit 3.3: Memory Units

1. Von Neumann Architecture

图示

描述已自动生成

1. Memory
   1. **Main memory: RAM, …**
   2. Secondary memory: disks, …
   3. Volatile / non-volatile

断电后，RAM中的内容被清除，但是硬盘上的内容仍然保存

1. **RAM (Random Access Memory)**
   1. Data
   2. Instruction
2. Perspective
   1. Physical
   2. **Logical**

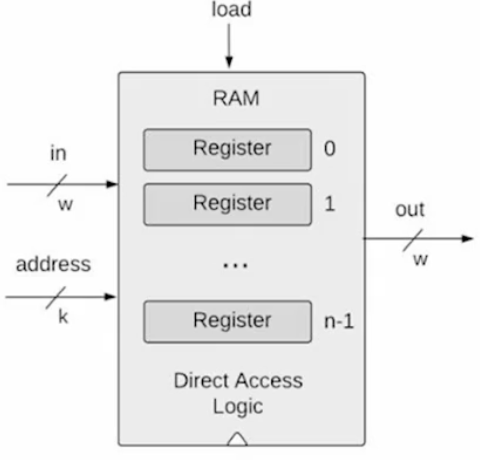
图示, 箭头

描述已自动生成

w (word width): **16-bit**(focus) 32-bit, 64-bit, …

Register’s state: the value which is currently stored inside the register

1. RAM unit

RAM abstraction:

A sequence of n addressable registers, with addresses 0 to n-1

At any given point of time, only one register in the RAM is selected

k (width of address input):

k = log 2 of n

w(word width): RAM is a sequential chip

No impact on the RAM logic with a clocked behavior

(Hack computer: w=16)

1. RAM Read Logic:

set address = i,

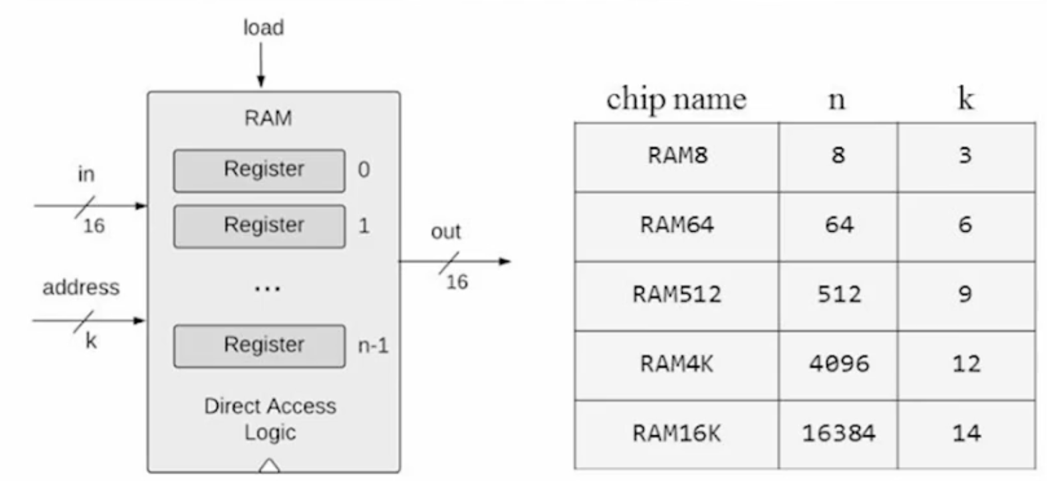
the “out” emits the state of Register i

RAM Write Logic:

set address = i, set in = v, set load = 1

the state of Register i becomes v, from the next cycle onward, “out” emits v

1. A family of 16-bit RAM chips



Unit 3.4: Counters

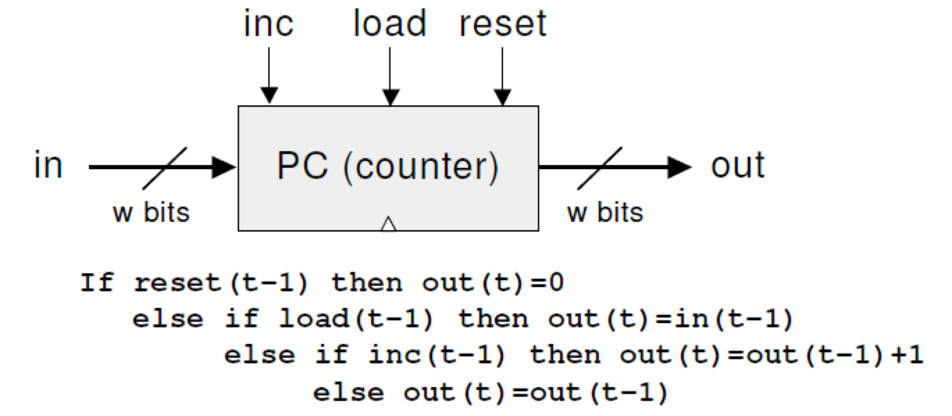
1. Three possible control settings:

Reset: fetch the first instruction PC = 0 // PC means Program Counter

Next: fetch the next instruction PC++

Goto: fetch instruction n PC = n

1. Counter: A chip that realizes this abstraction



图示

描述已自动生成