【CA】Day7

Course	Nand to Tetris
Study Date	@February 21, 2022

【Ch3】Sequential Logic

All the Boolean and arithmetic chips that we built in chapters 1 and 2 were combinational. Those chips can compute functions that depend solely on combinations of their input values, but they cannot maintain state.

They must be equipped with memory elements that can preserve data over time. These memory elements are built from sequential chips.

3.1 Background

In order to build chips that “remember” information, we must first develop some standard means for representing the progression of time.

- The Clock: In most computers, the passage of time is represented by a master clock that delivers a continuous train of alternating signals. The exact hardware implementation is usually based on an oscillator that alternates continuously between two phases labelled 0-1, low-high, tick-tock.

  The elapsed time between the beginning of a “tick” and the end of the subsequent “tock” is called cycle, and each clock cycle is taken to model one discrete time unit.

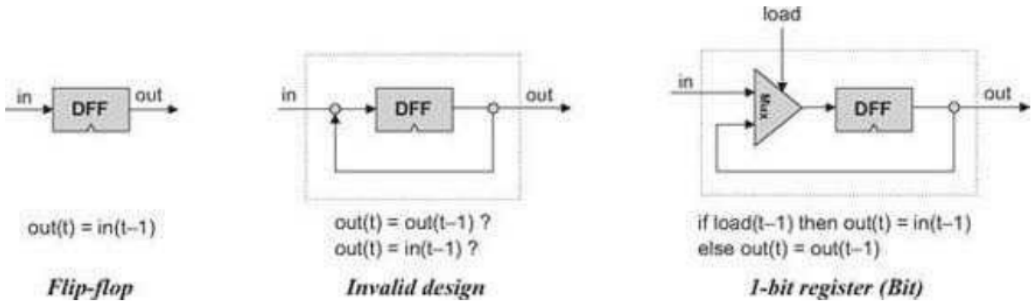
- Flip-Flops: The most elementary sequential element in the computer is a device called a flip-flop, of which there are several variants.

  In this book we use a variant called a data flip-flop, or DFF, whose interface consists of a single-bit data input and a single-bit data output.

  In addition, the DFF has a clock input that continuously changes according to the master clock’s signal. Taken together, the data and the clock inputs enable the DFF to implement the time-based behaviour out(t) = in(t-1), where in and out are the gate’s input and output values and t is the current clock cycle.

- Registers: A register is a storage device that can store a value over time, implementing the classical storage behaviour out(t)=out(t-1).

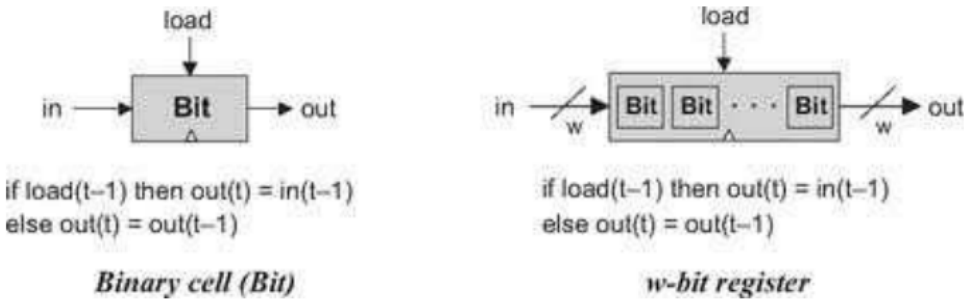
  A DFF, on the other hand, can only output its previous input, out(t) = in(t-1). This suggests that a register can be implemented from a DFF by simply feeding the output of the latter back into its input.



The rightmost figure shows the correct implementation of a register. We can combine a multiplexor with a DFF.

The “select bit” of this multiplexor can become the “load bit” of the overall register chip: If we want the register to start storing a new value, we can put this value in the input and set the load bit to 1; if we want the register to keep storing its internal value until further notice, we can set the load bit to 0.

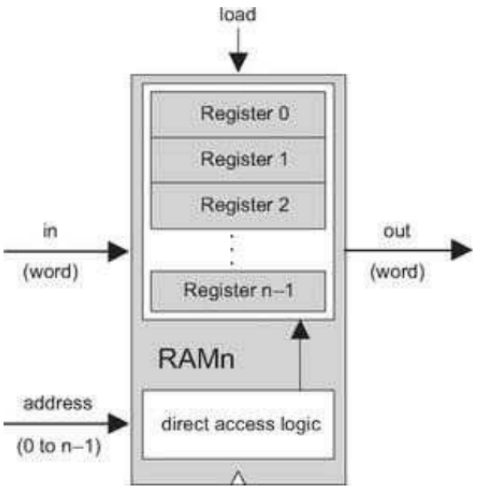
Once we have developed the basic mechanism for remembering a single bit over time, we can easily construct arbitrarily wide registers. This can be achieved by forming an array of as many single-bit registers as needed, creating a register that holds multi-bit values. The basic design parameter of such a register is its width-the number of bits that it holds-e.g., 16, 32, or 64. The multi-bit contents of such registers are typically referred to as words.



- Memories: We can now proceed to build memory banks of arbitrary length. As the following figure shows, this can be done by stacking together many registers to form a Random Access Memory RAM unit.

  The term random access memory derives from the requirements that read/write operations on a RAM should be able to access randomly chosen words, with no restrictions on the order in which they are accessed.

  That is to say, we require that any word in the memory be accessed directly, in equal speed.



re 3.3 RAM chip (conceptual). The width and length of the RAM can vary.

The requirement can be satisfied as follows:

1. First, we assign each word in the n-register RAM a unique address(an integer between 0 to n-1), according to which it will be accessed.
2. Second, in addition to building an array of n registers, we built a gate logic design that, given an address j, is capable of selecting the individual register whose address is j.

In sum, a classical RAM device accepts three inputs: a data input, an address input, and a load bit. The address specifies which RAM register should be accessed in the current time unit.

In the case of a read operation(load = 0), the RAM's output immediately emits the value of the selected register.

In the case of a write operation(load = 1), the selected memory register commits to the input value in the next time unit, at which point the RAM's output will start emitting it.

The basic design parameters of a RAM device are its data width-the width of each one of its words, and its size-the number of words in the RAM.

- Counters: A counter is a sequential chip whose state is an integer number that increments every time unit, effecting the function out(t) = out(t - 1) + c, where c is typically 1.

  Counters play an important role in digital architectures.

  For example, a typical CPU includes a program counter whose output is interpreted as the address of the instruction that should be executed next in the current program.