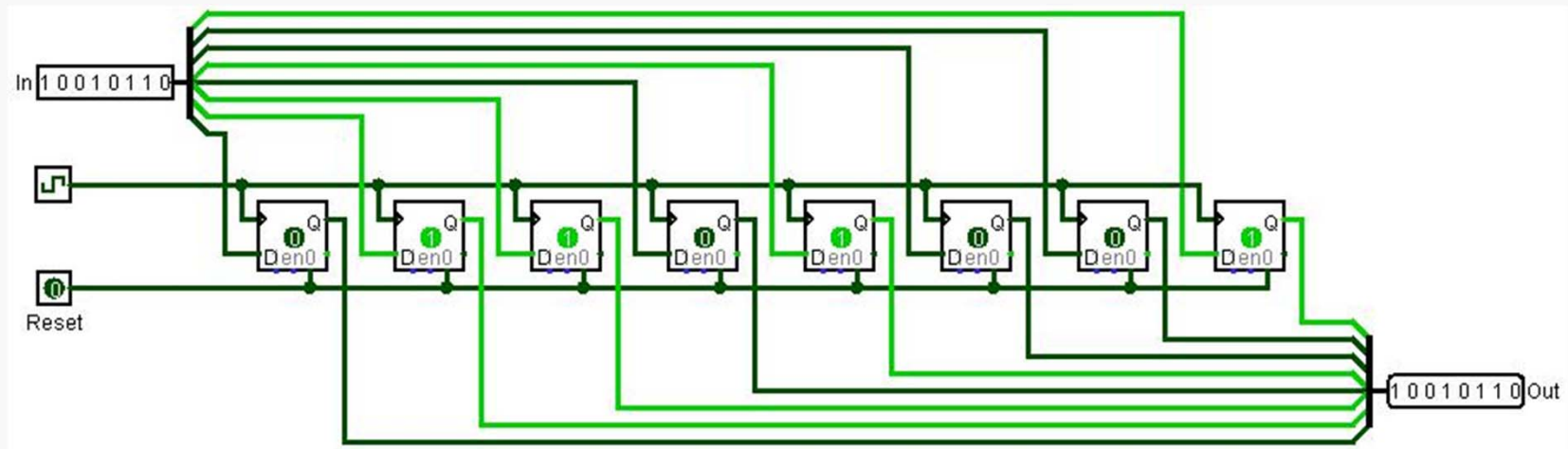


# 8-Bit Register

Built using D flip-flops:



Clock input controls when input is "written" to the individual flip-flops.

However, the design above isn't quite what we want...

**QTP What's wrong with this?  
How can we fix it?**

A *register file* is a collection of  $k$  registers (a sequential logic block) that can be read and written by specifying a register number that determines which register is to be accessed.

The interface should minimally include:

- an  $n$ -bit input to import data for writing (a *write port*)
- an  $n$ -bit output to export read data (a *read port*)
- a  $\log(k)$ -bit input to specify the register number
- control bit(s) to enable/disable read/write operations
- a control bit to clear all the registers, asynchronously
- a clock signal

Some designs may provide multiple read or write ports, and additional features.

For MIPS, it is convenient to have two read ports and one write port. Why?

# A File of 8-Bit Registers

Aggregating a collection of 4-bit registers, and providing the appropriate register selection and data input/output interface:

