6.823 Computer System Architecture

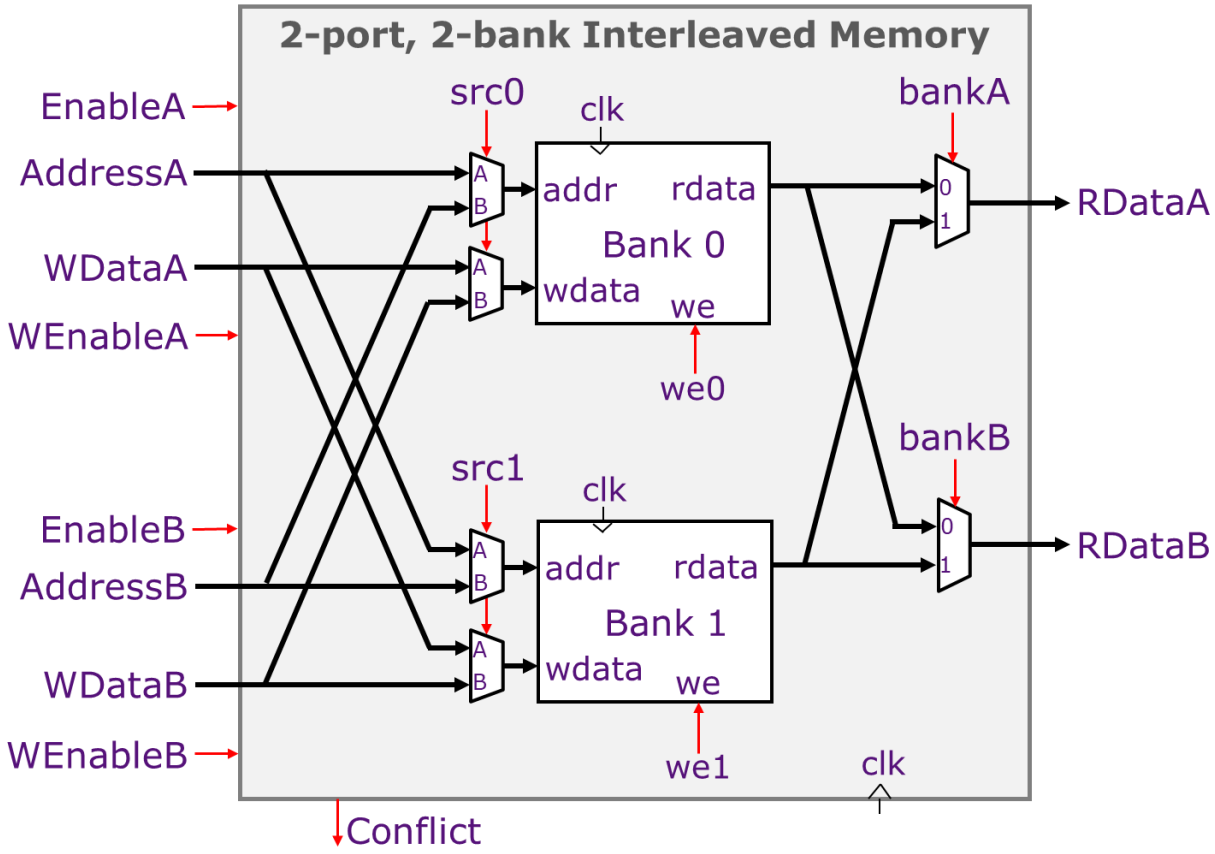
Interleaved Memory

http://csg.csail.mit.edu/6.823/

Multiported memories are expensive: for each port to independently read or write any location in the memory, each port requires its own bitlines and wordlines. Thus, the silicon area needed grows quadratically with the number of ports. *Banking* is a popular approach to increase memory throughput without such high hardware costs. In a *banked memory*, the data is partitioned into smaller *memory banks*, each of which uses a single-port memory. Accesses to data in different banks can be done in parallel, whereas accesses to the same bank must be serialized.

When dividing memory into banks, one must make a design decision for how to place data across the banks. A common choice is to interleave consecutive words across banks, so that scanning through a contiguous range of addresses would access each bank in turn. A memory divided into banks in this way is called an *interleaved memory*.

Consider an memory where data is interleaved across two banks at the granularity of 4-byte words. The interleaved memory is implemented using a single-cycle memory for each bank.



Bank 0 holds all *even* words of memory, i.e., the words at addresses 0x0, 0x8, 0x10, 0x18, etc. Bank 1 holds all *odd* words of memory, i.e., the words at addresses 0x4, 0xC, 0x14, 0x1C, etc.

Assume each bank performs combinational reads, so reads happen within a single cycle, while writes are performed at the end of a clock cycle.

The interleaved memory has two external ports, A and B. Each port can access up to one 4-byte word per cycle. Each port has an enable signal which indicates whether an access is requested on that port. Each port also has a write-enable signal which indicates if the address being requested should be written to. If an access is performed for port A, the data will always be sent to RDataA, and likewise if an access is performed for port B, the data will be sent to RDataB.

A and B can be used in the same cycle only if they access different banks. If access are requested by both ports in a single cycle and the requested addresses belong to the same bank, this is called a *bank conflict*, and only one of the two ports can have its access performed. In this case, the interleaved memory will assert the 1-bit output Conflict.

In the event of a bank conflict, this interleaved memory prioritizes port A. If EnableA is asserted, port A is guaranteed to have its access performed, and any request on port B will only be accessed if there is no bank conflict. If EnableA is not asserted, then no action is performed for port A, and port B's access is performed.

In order to implement the behavior described above, the interleaved memory computes Conflict and its internal control signals (for multiplexers and each bank's write-enable) using combinational logic, based on the external inputs to the interleaved memory.