Using MML to Simulate Dual-Ported SRAMs

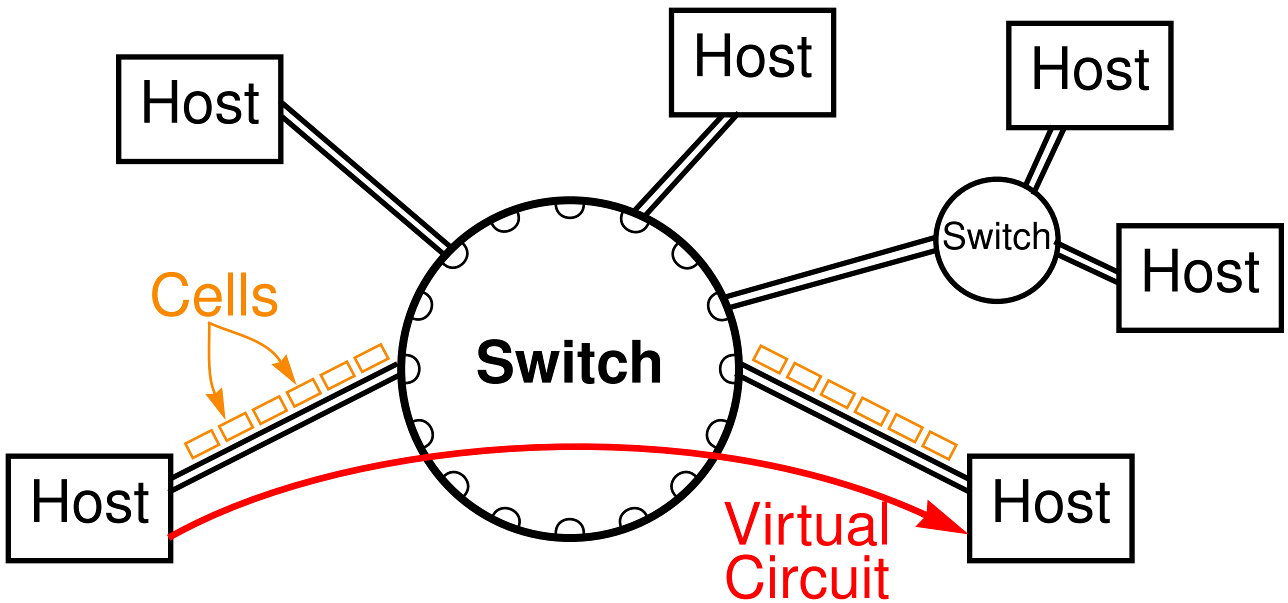
Parallel Routing Lookups in an ATM Switch Controller

Aaron Brown, Dan Chian, Nishat Mehta, Yannis Papaefstathiou, Josh Simer, Trevor Blackwell, Michael D. Smith, Woodward Yang
Harvard University

ISCA Workshop on Mixing Logic and DRAM
1 June 1997

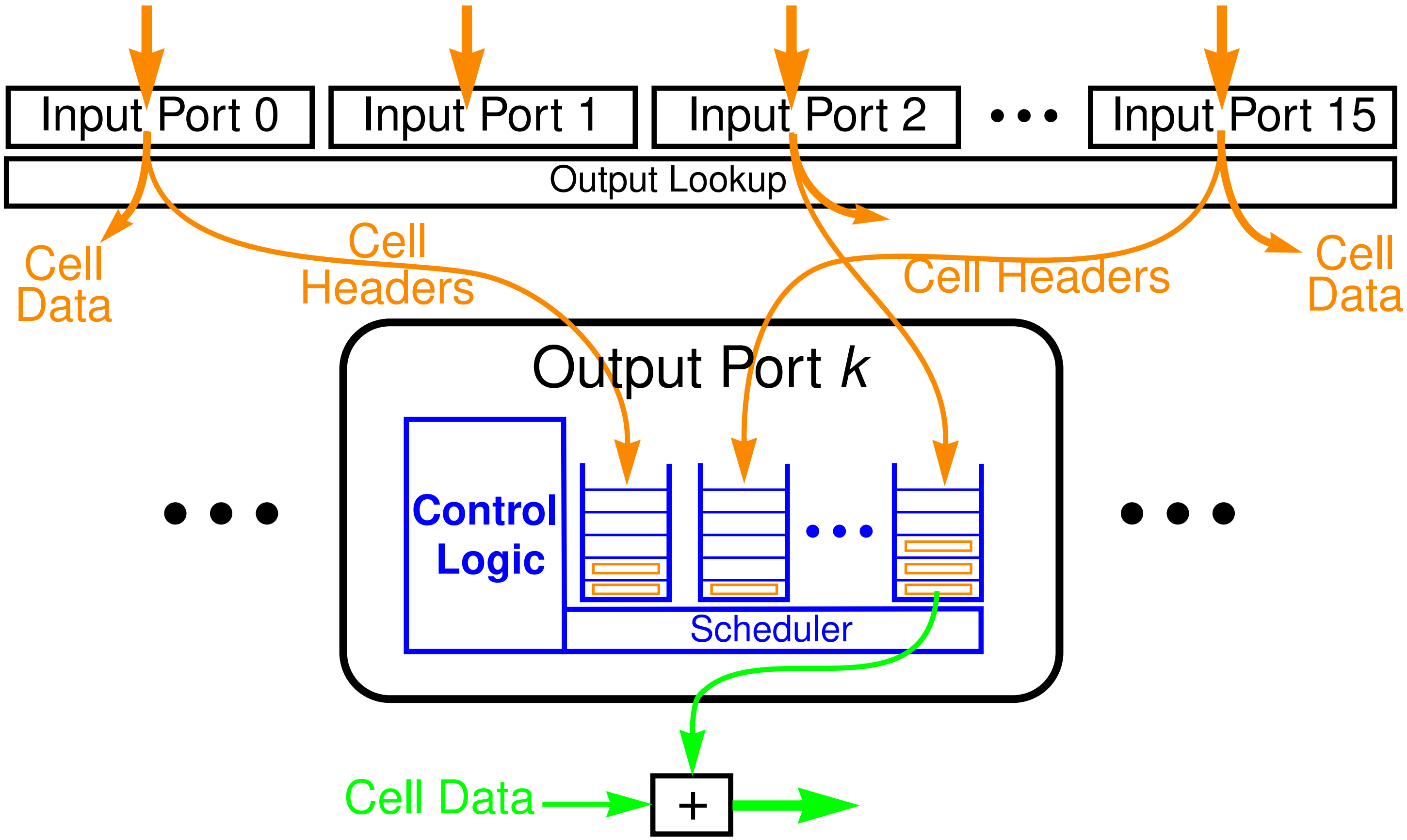
The Problem: ATM Switching

- **ATM network:**

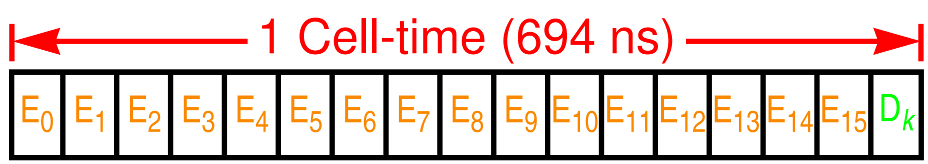


- **Switch Example: Harvard CreditSwitch**
  - 16 bidirectional 622 Mb/sec ports (OC-12)
  - 10 Gb of aggregate throughput per second
  - Up to 46 million cells processed per second

Inside an ATM Switch

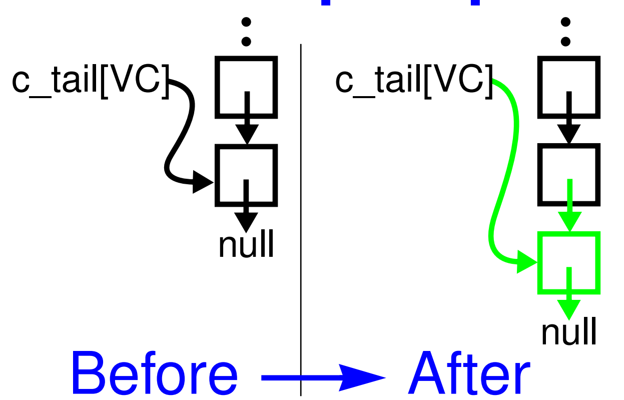


Output Port Timing:



Output Port Control Details

- **Example operation: *cell enqueue***




```
cell_enqueue(VC, Caddr) {
    c_link[c_tail[VC]] = Caddr;
    c_tail[VC] = Caddr;
    /* ... */
}
```

Pseudocode

- **Lists implemented as hardware tables**
  - All accesses are in the form *Read-Modify-Write*
  - Access time to tables must be < 20.4 ns
- **Total table size for all output ports is 20 Mb**
  - CreditSwitch requires 272 20ns dual-ported SRAMs to implement the output ports

MML for Integration

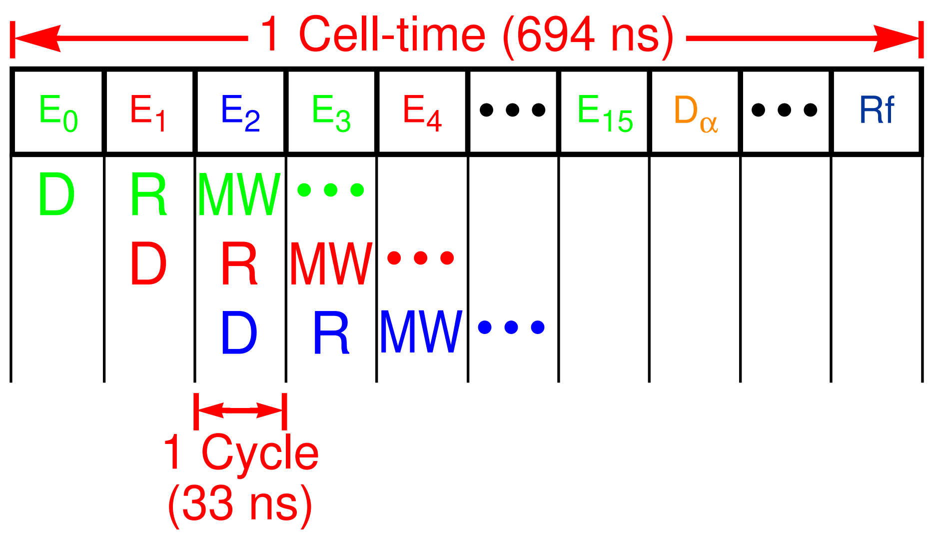
- **Too much memory to integrate in SRAM**
- **Standard DRAM has the capacity but not the speed, and needs refresh**
- **MML is the solution**
  - Combining logic and memory reduces access-time demands
  - Allows modification of DRAM block to simulate dual-porting

An MML Implementation

- **Issues**
  - Timing
  - Organization
- **Read-Modify-Write (RMW) is the key**
  - Not limited to an ATM switch
  - Useful anywhere linked lists are manipulated in hardware

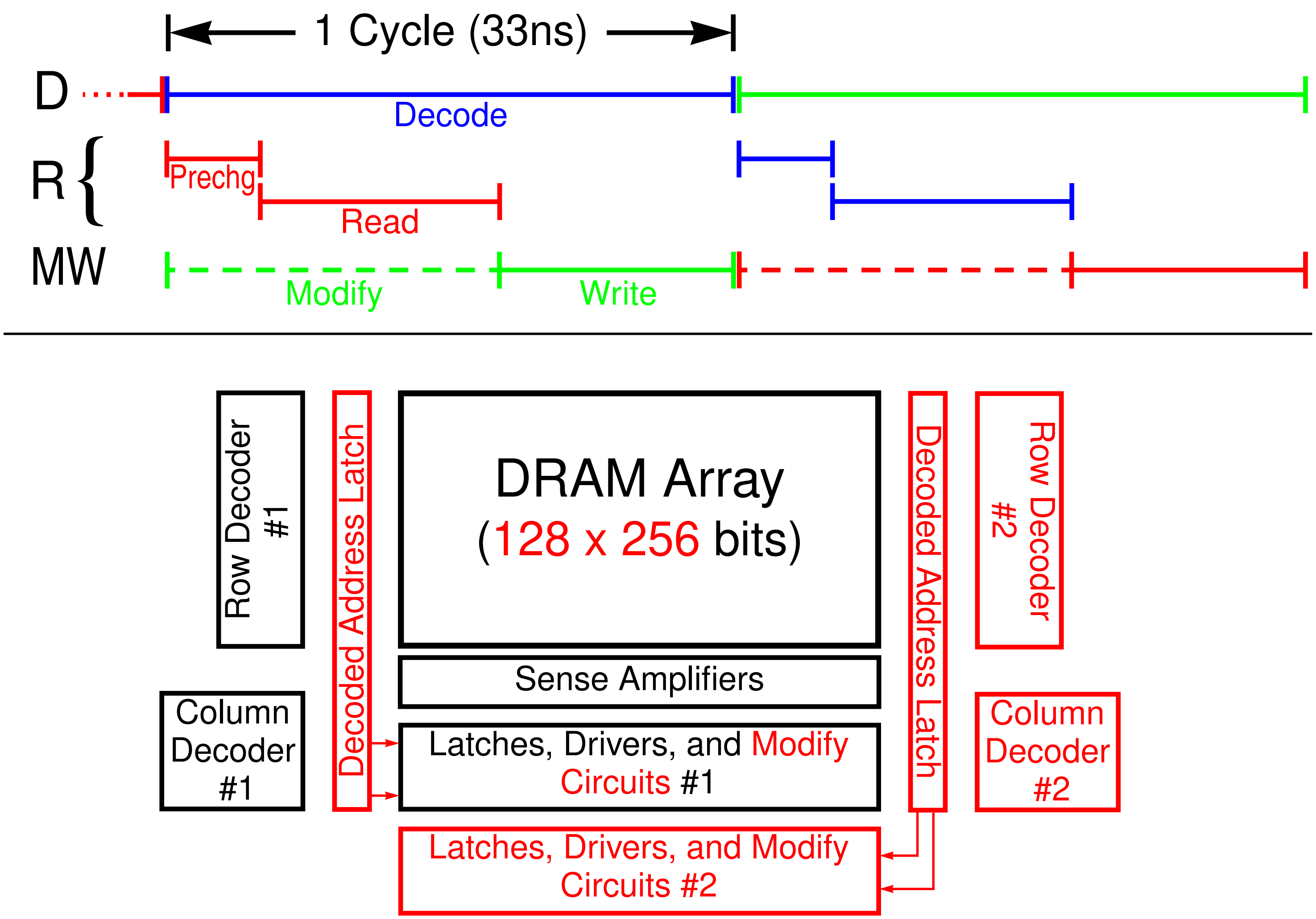
General Timing

- **Divide each RMW operation into 3 cycles: decode (D), read (R), and modify/write (MW)**



- **Use spare cycles in the pipeline for refresh**

Memory Timing/Organization



Timing Feasibility

- **Comparison with 64Mb DRAM**

Time Component	64 Mbit DRAM	Our MML Design
Drive word lines for read	9 ns	4 ns
Precharge bit lines for read	15 ns	7 ns
Precharge bit lines for write	15 ns	0 ns
Drive word lines for write	9 ns	4 ns
Drive bit lines for write	25 ns	8 ns
Sense amplifiers	9 ns	9 ns

- **Implementation tricks:**
  - Remove precharge for write
  - Take advantage of implicit write following a read
  - Duplicate decoders and latches to allow pipelining
  - Latch decoder outputs for quick reuse

Conclusions

- **Fast RMW is possible in DRAM technology**
  - Provides characteristics of dual-ported SRAM
- **It is useful to optimize DRAM for aspects other than** density
  - Speed
  - Functionality
- **Real application: ATM switching**
  - Cheaper “smart” switches
  - Applicability is wider