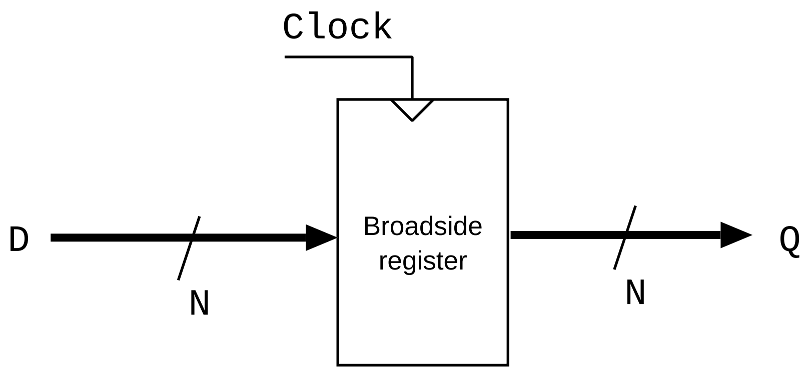
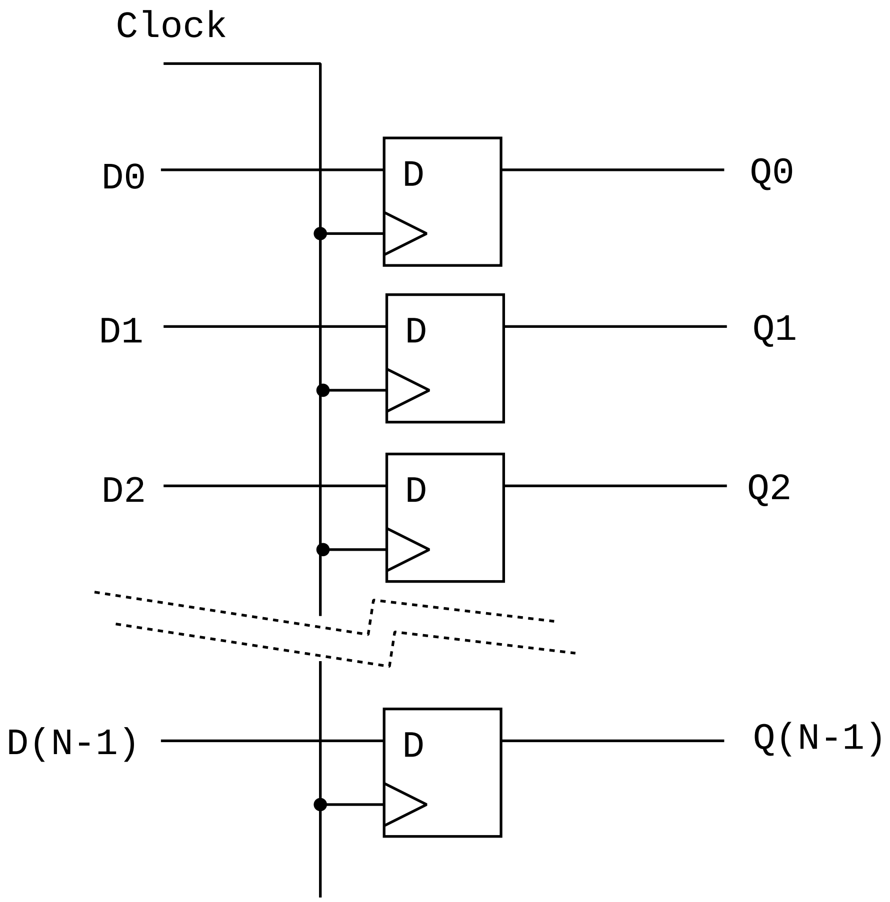
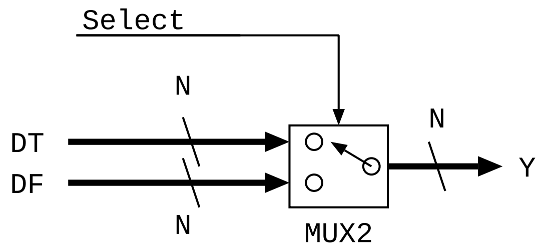
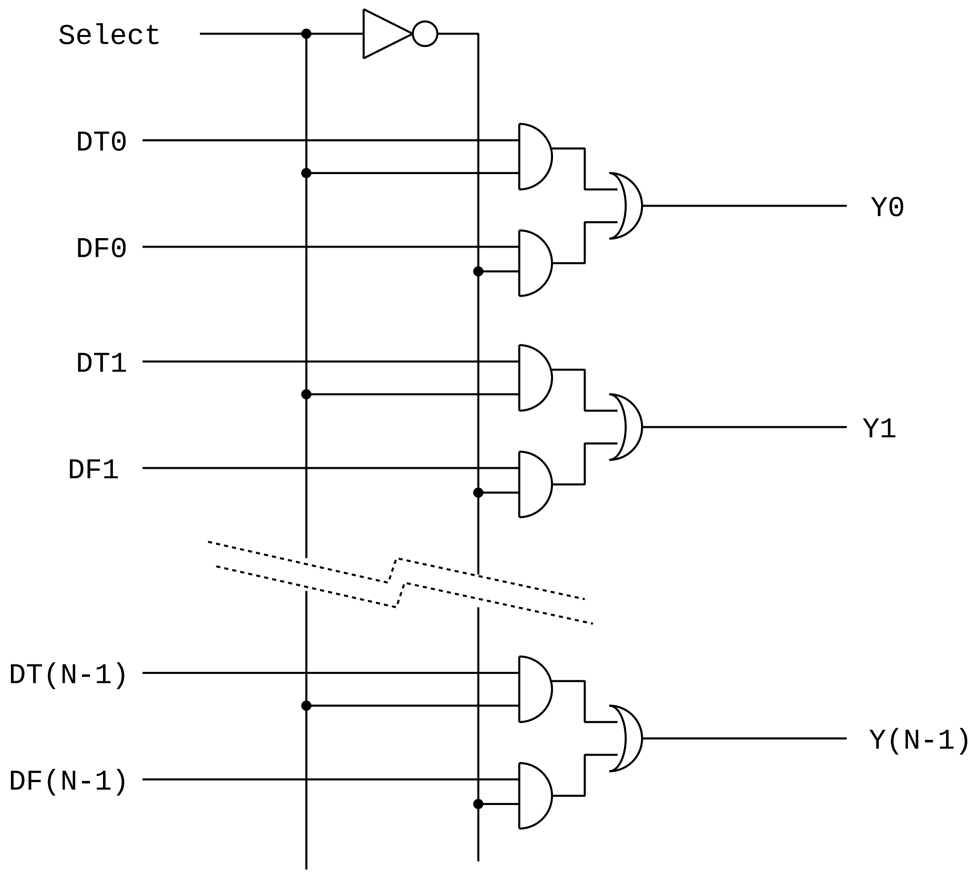
A Broadside Register



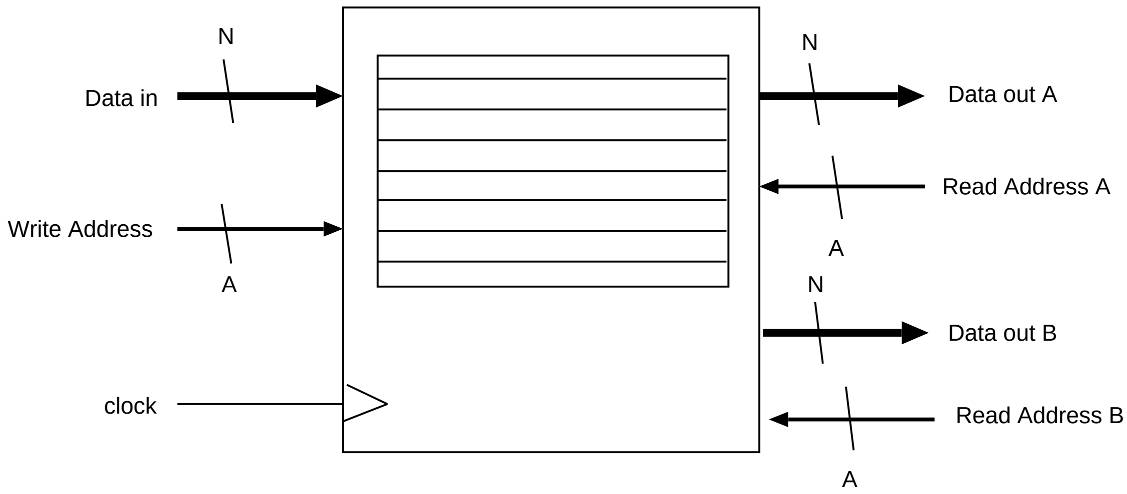


A broadside two-to-one multiplexor

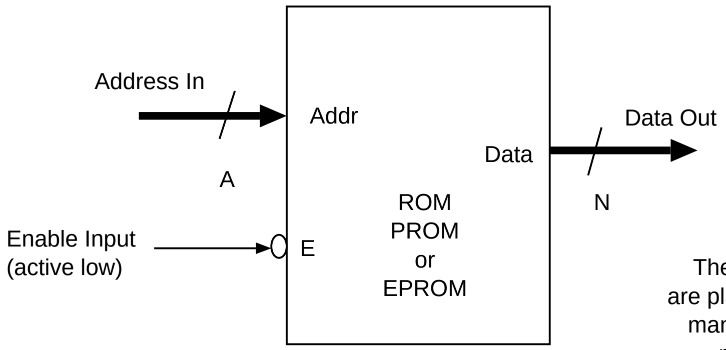




A dual port register file

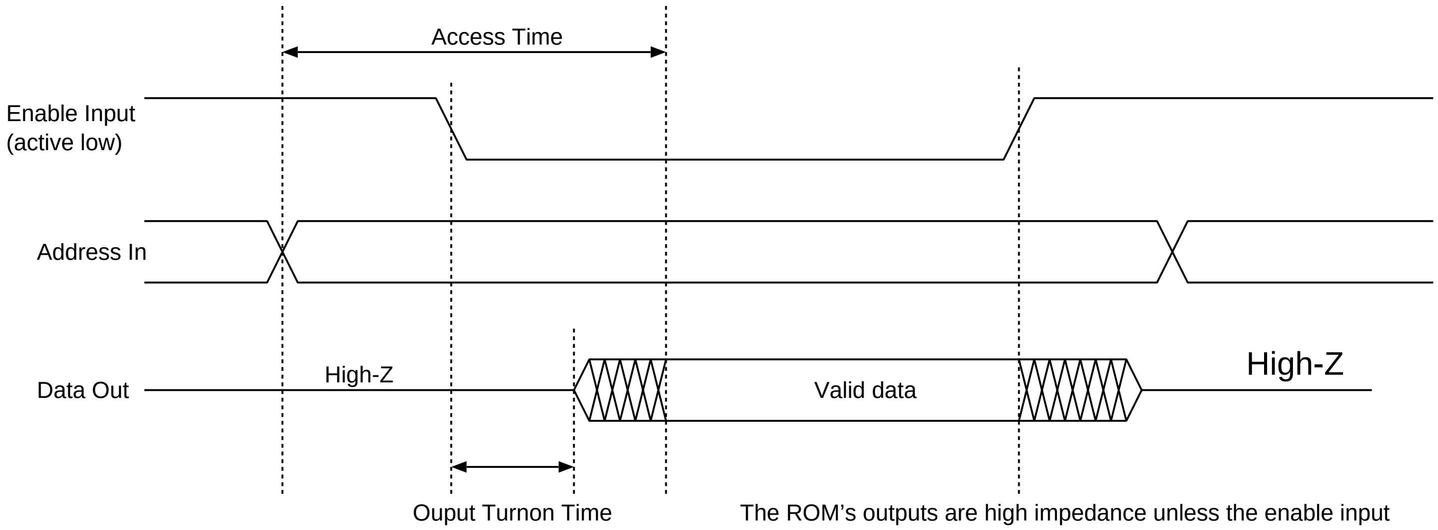


Read Only Memory (ROM)

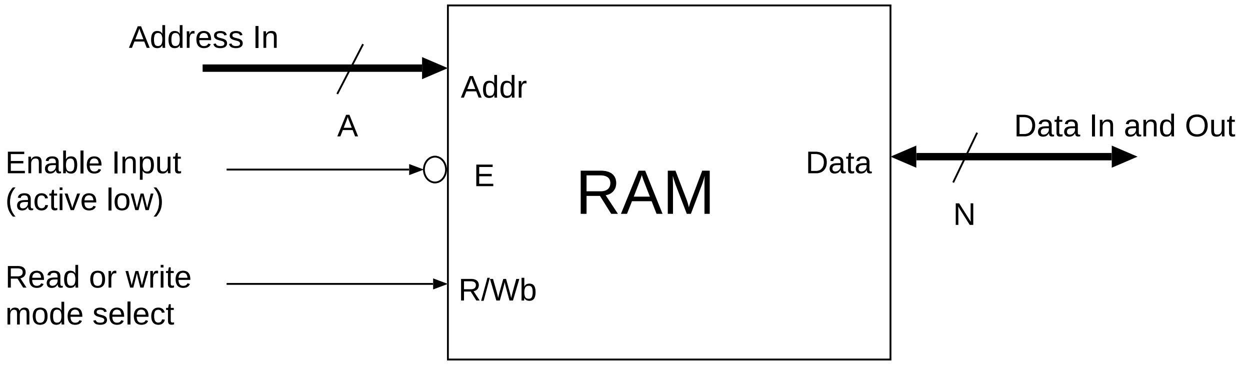


The ROM contents are placed inside during manufacture or field programming.

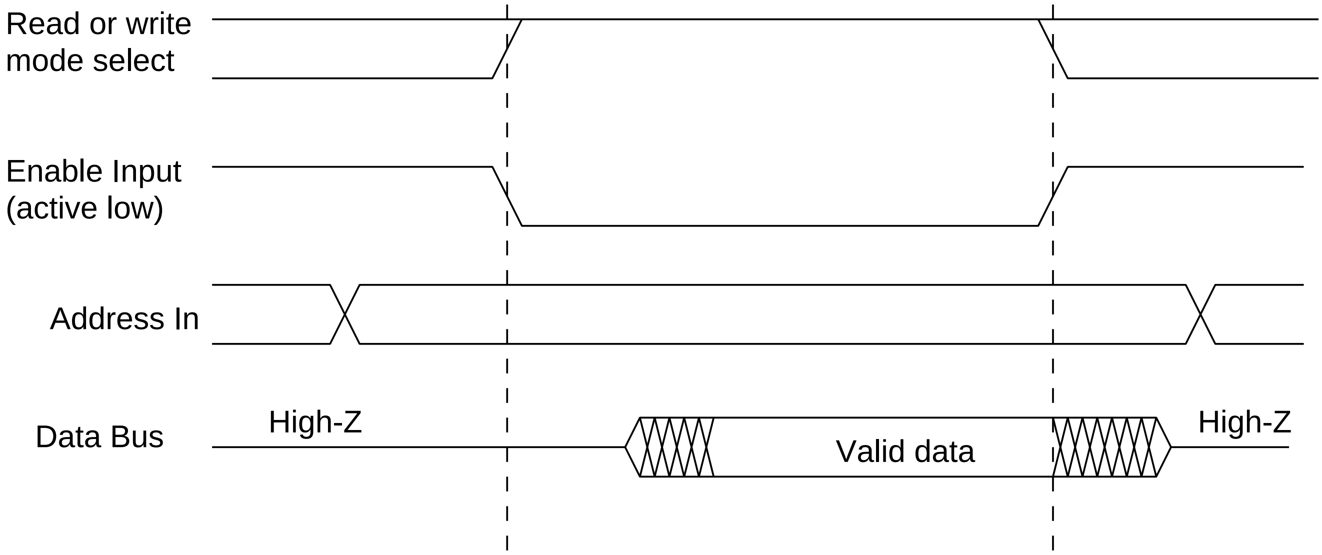
The ROM takes A address bits named A0 to A<A-1> and produces data words of N bits wide. For example, if A=5 and D=8 then the ROM contains 2**5 which is 32 locations of 8 bits each. The address lines are called A0, A1, A2, A3, A4 and the data lines D0, D1, ... D7



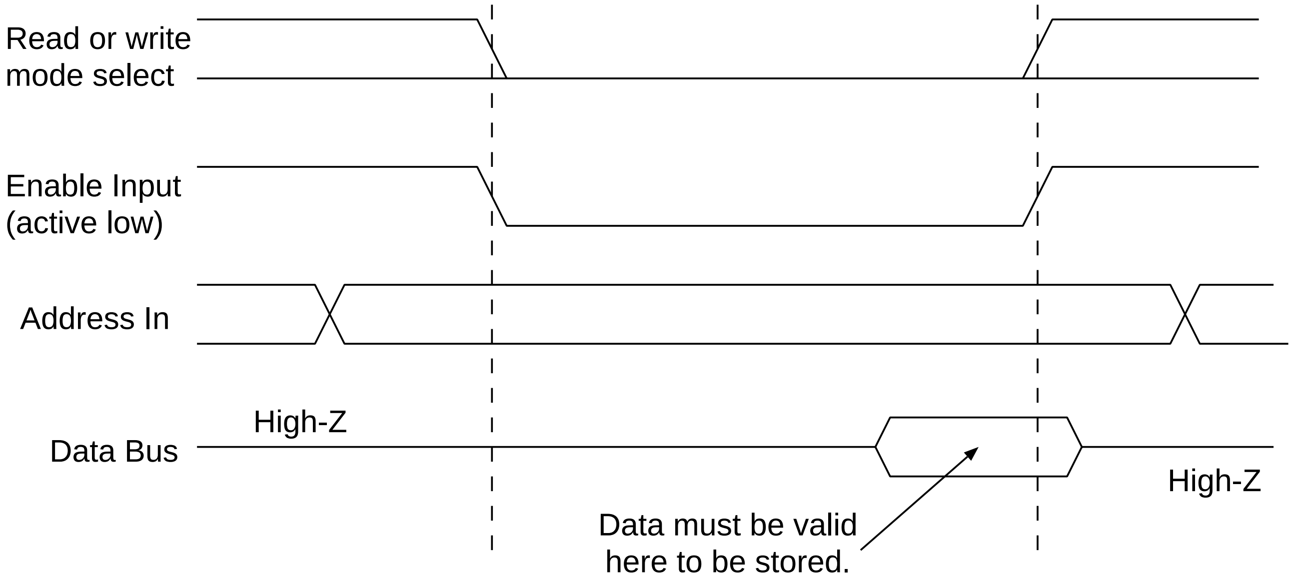
The ROM's outputs are high impedance unless the enable input is asserted (low). After the enable is low the output drivers turn on. When the address has been stable sufficiently long, valid data from that address comes out.



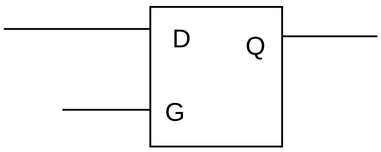
Read Cycle - Like the ROM



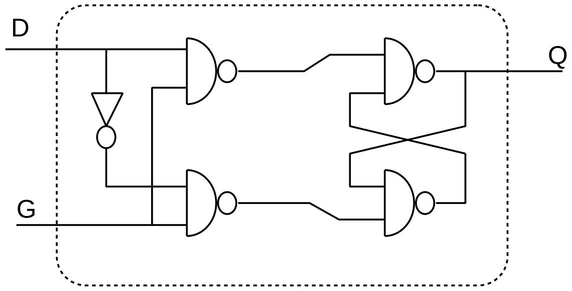
Write Cycle - Data stored internally



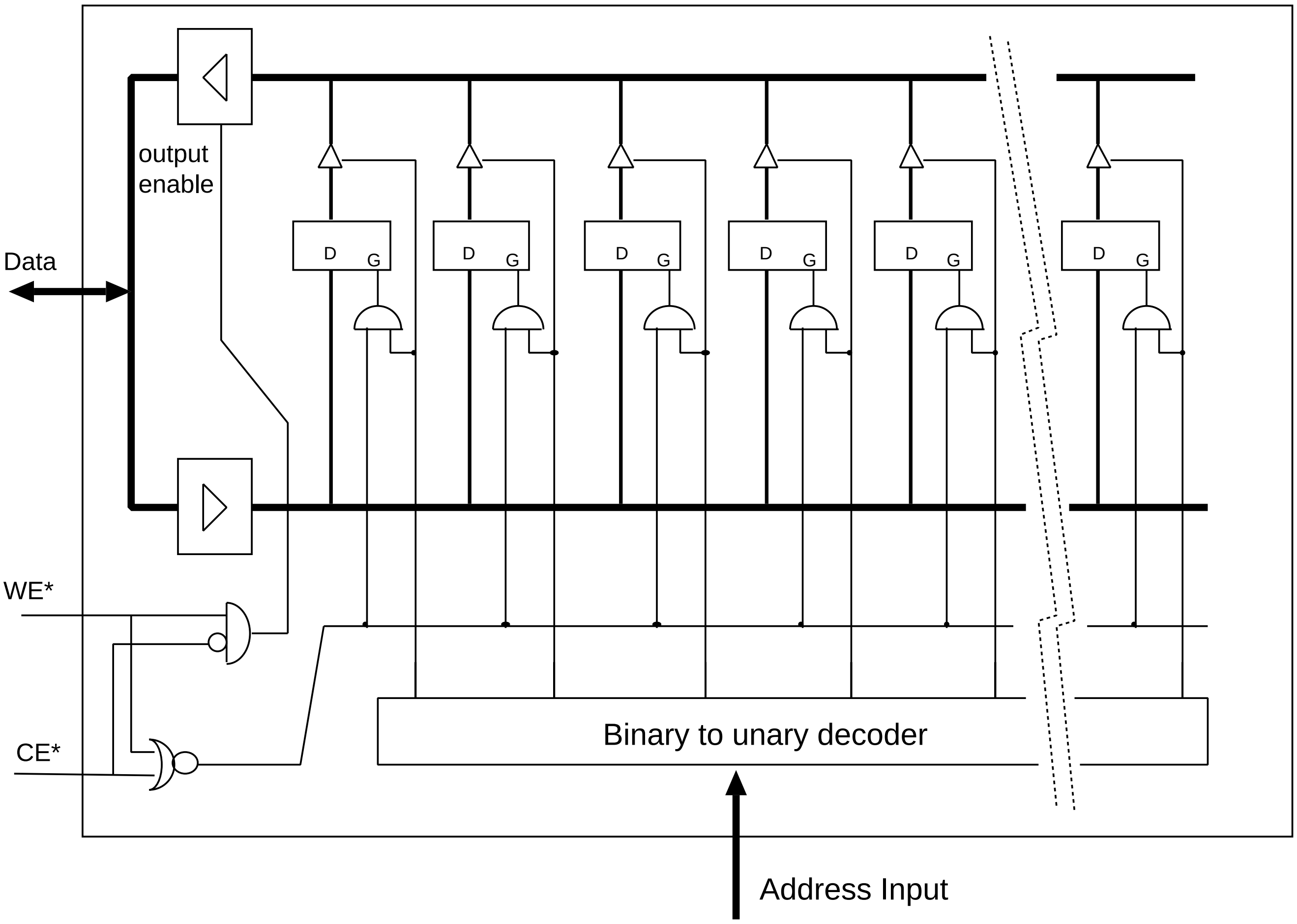
Unlike the edge-triggered flip-flop, the transparent latch passes data through in a transparent way when its enable input is high. When its enable input is low, the output stays at the current value.

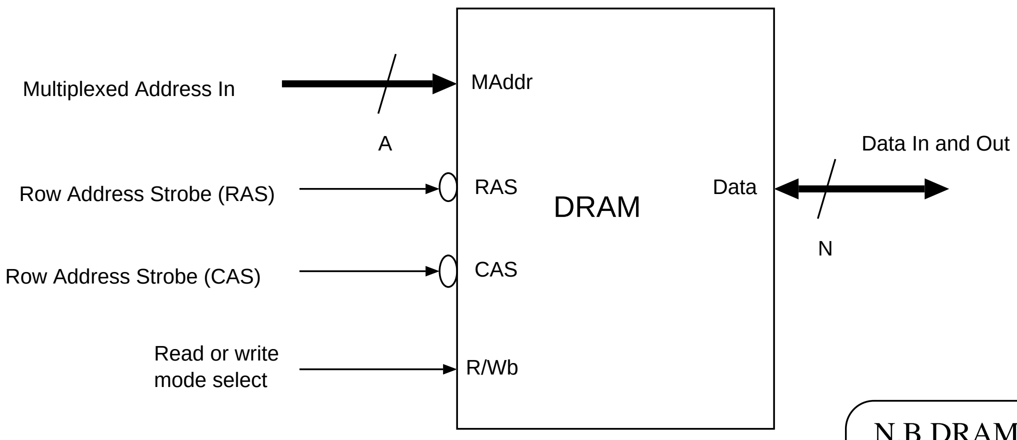


Transparent latch schematic symbol



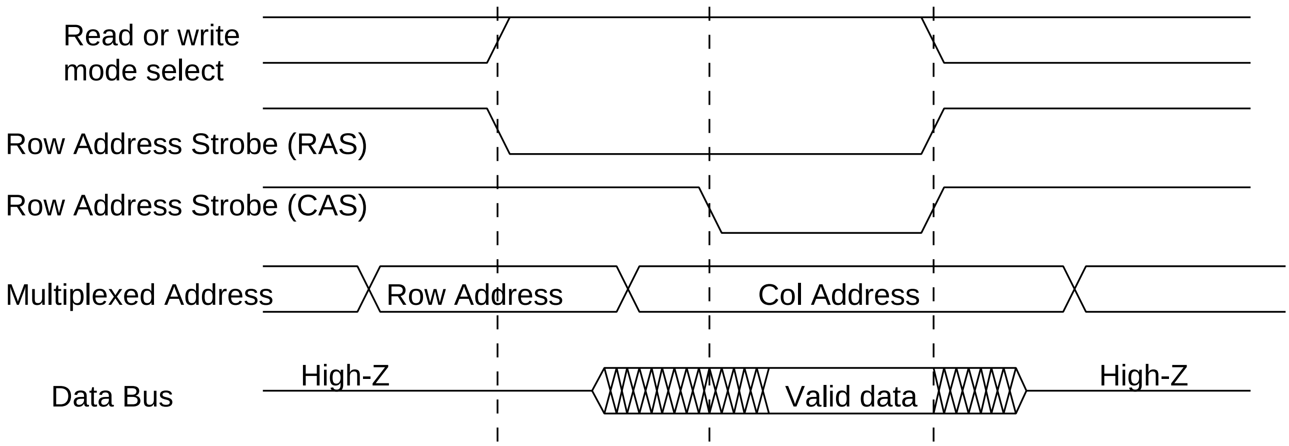
Transparent latch implemented from gates.





N.B DRAM will not be examinable for 1A.

Read Cycle (write is similar)

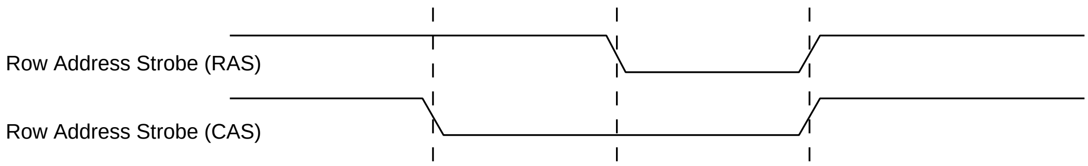


A DRAM has a multiplexed address bus and the address is presented in two halves, known as row and column addresses. So the capacity is 4**A x D. A 4 Mbit DRAM might have A=10 and D=4.

When a processor (or its cache) wishes to read many locations in sequence, only one row address needs be given and multiple col addresses can be given quickly to access data in the same row. This is known as 'page mode' access.

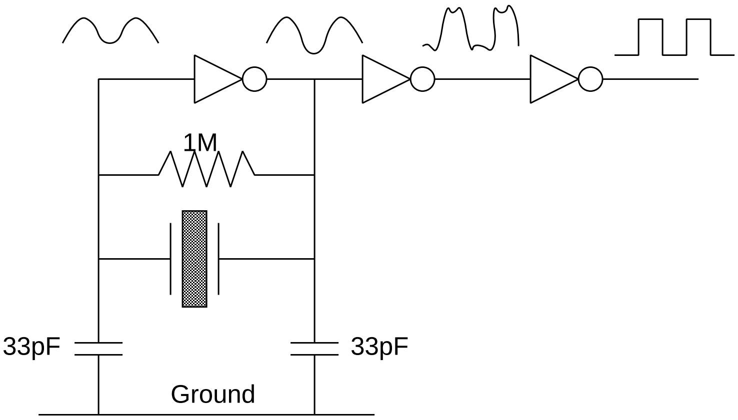
EDO (extended data out) DRAM is now quite common. This guarantees data to be valid for an exteneded period after CAS, thus helping system timing design at high CAS rates.

Refresh Cycle - must happen sufficiently often!

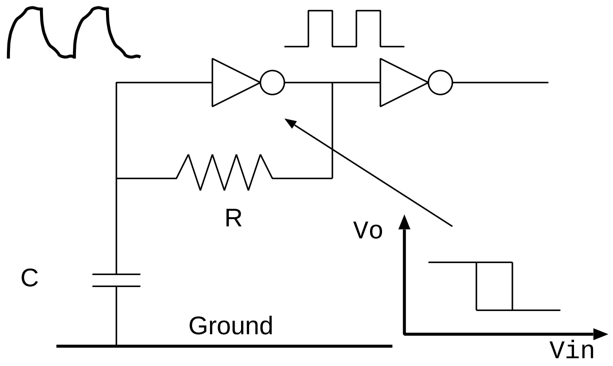


No data enters or leaves the DRAM during refresh, so it 'eats memory bandwidth'.
Typically 512 cycles of refresh must be done every 8 milliseconds.

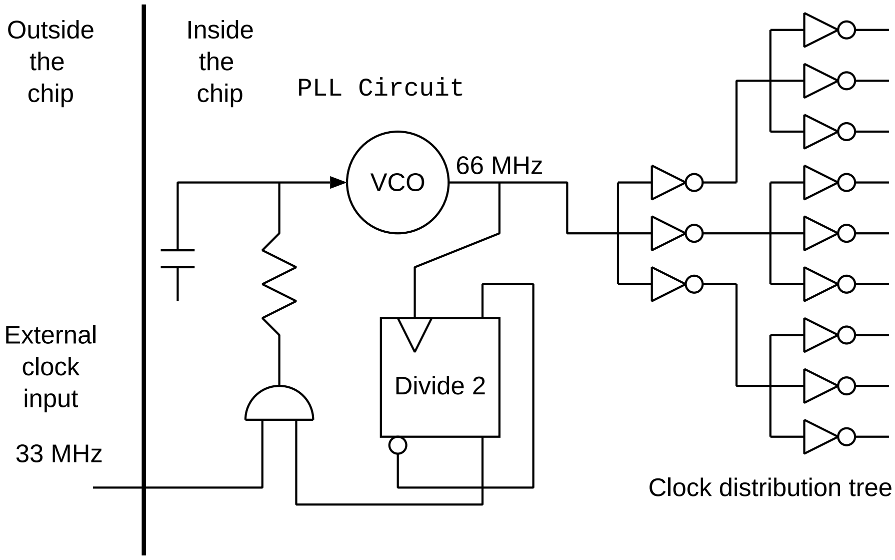
Crystal oscillator clock source



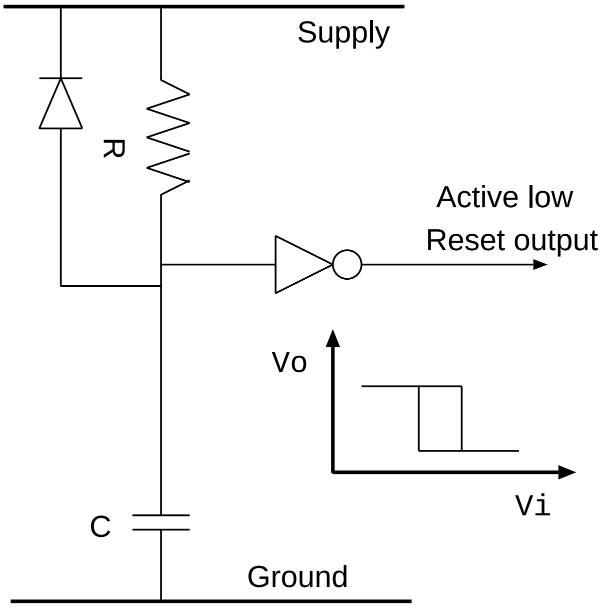
RC oscillator clock source



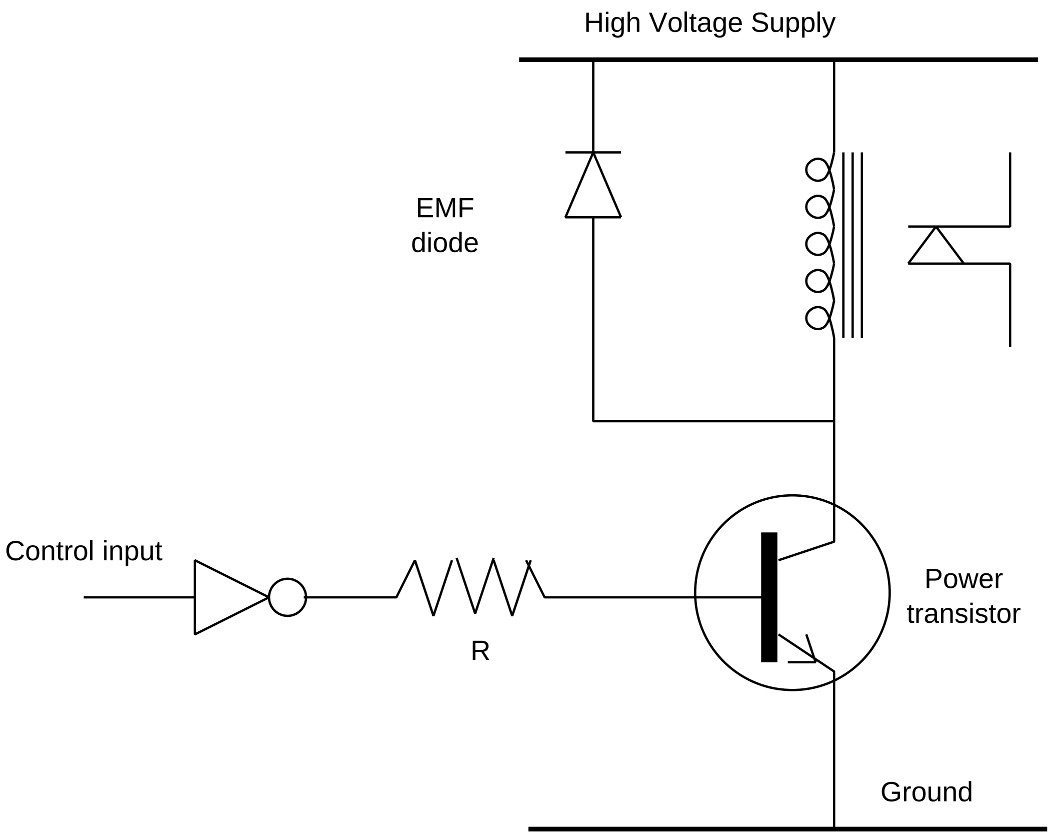
Clock multiplication and distribution



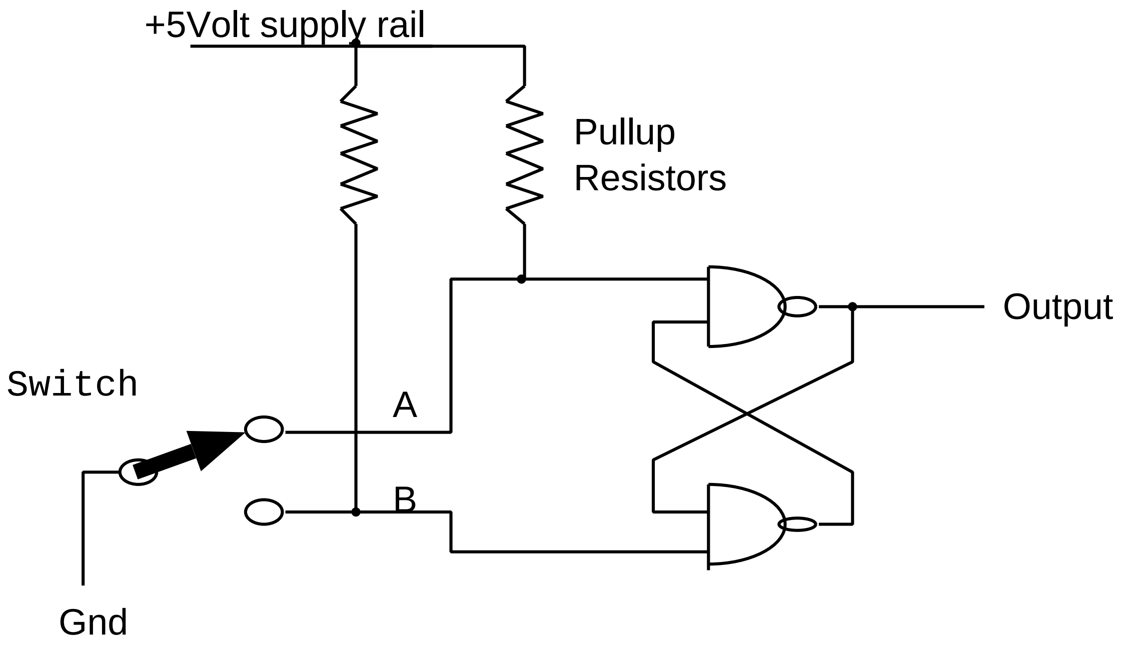
Power-on reset

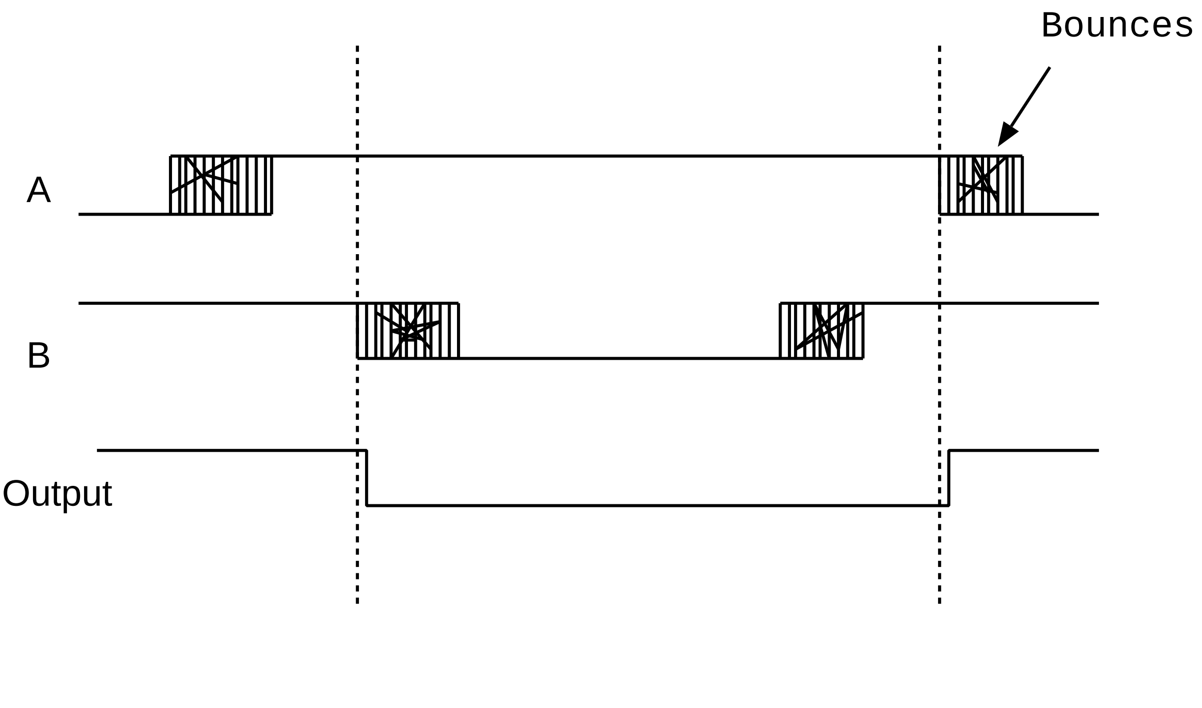


Driving a heavy current or high-voltage load

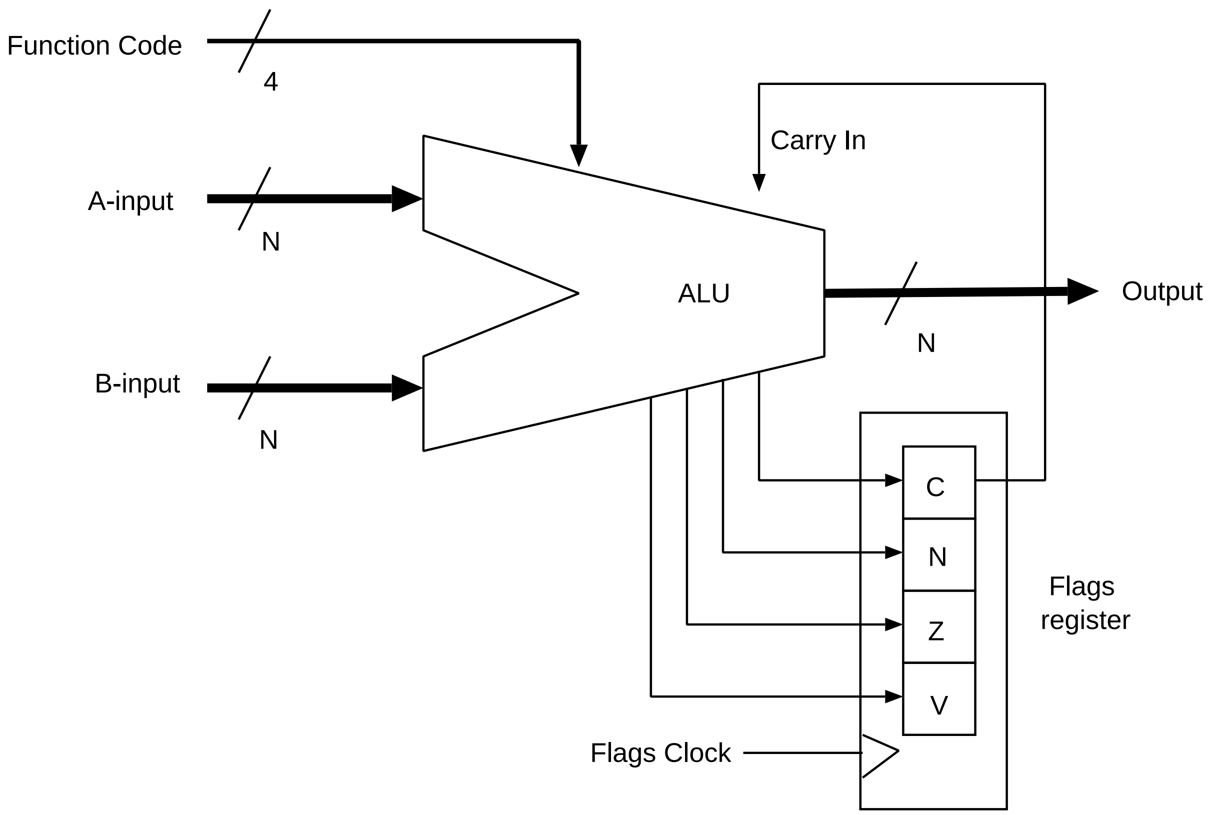


Debouncer circuit for a two-pole switch

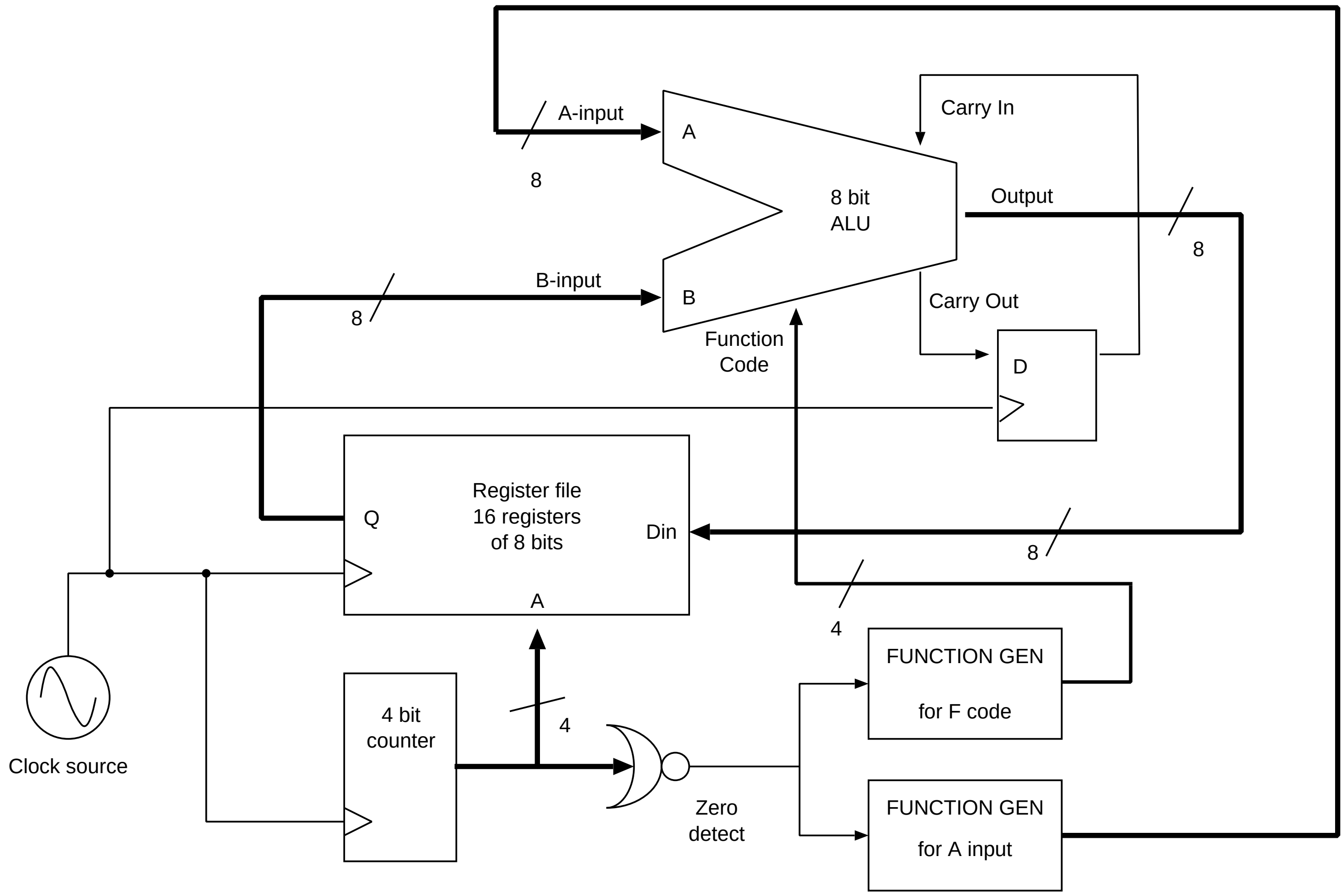




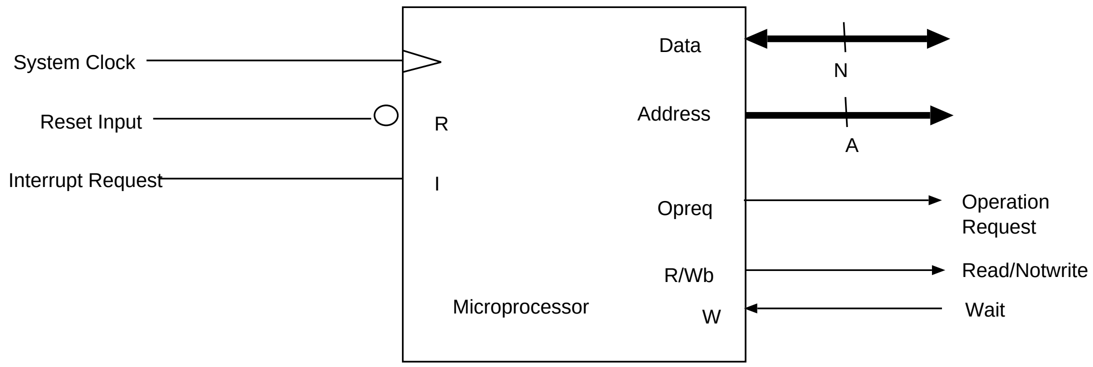
ALU and flags register



ALU and register file

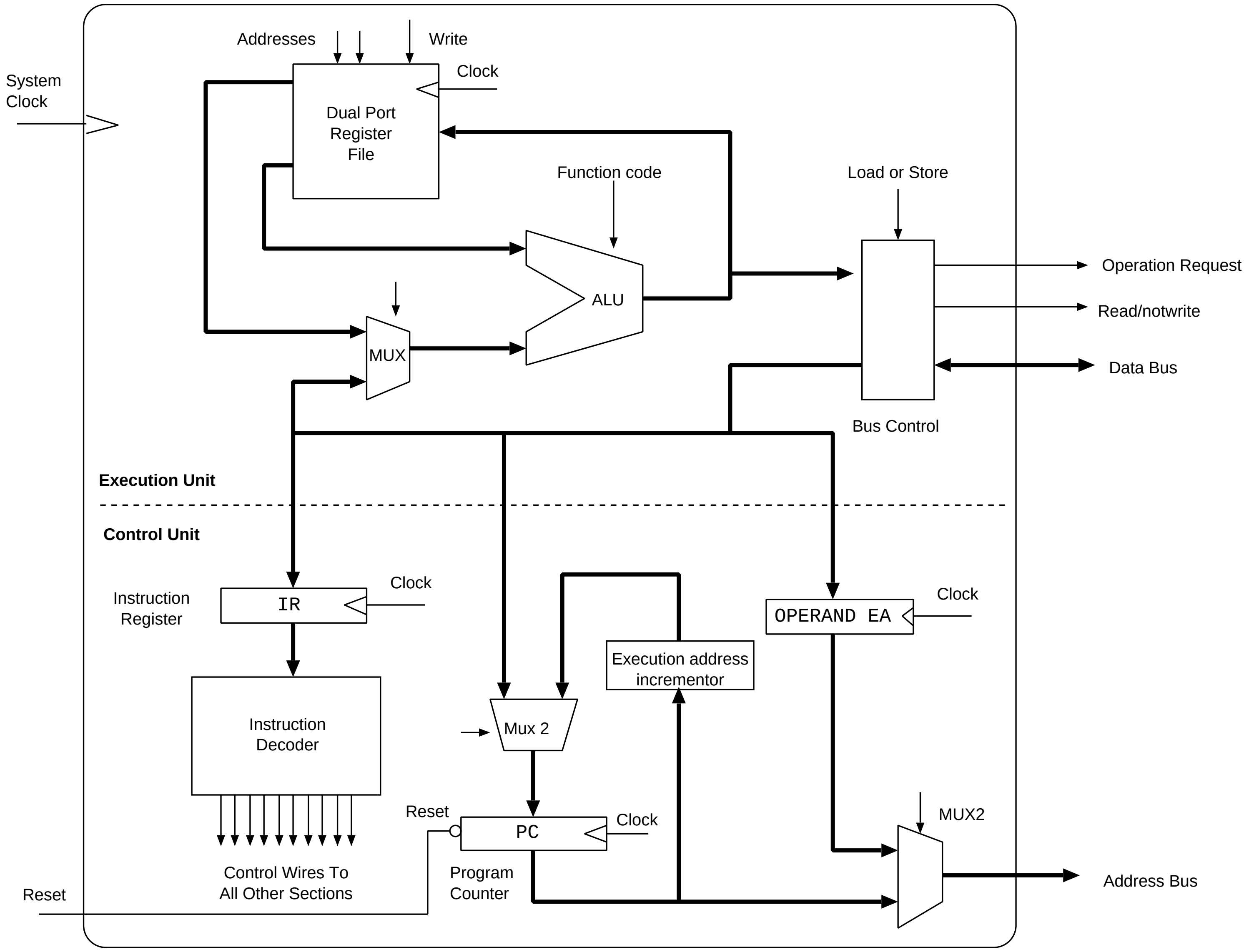


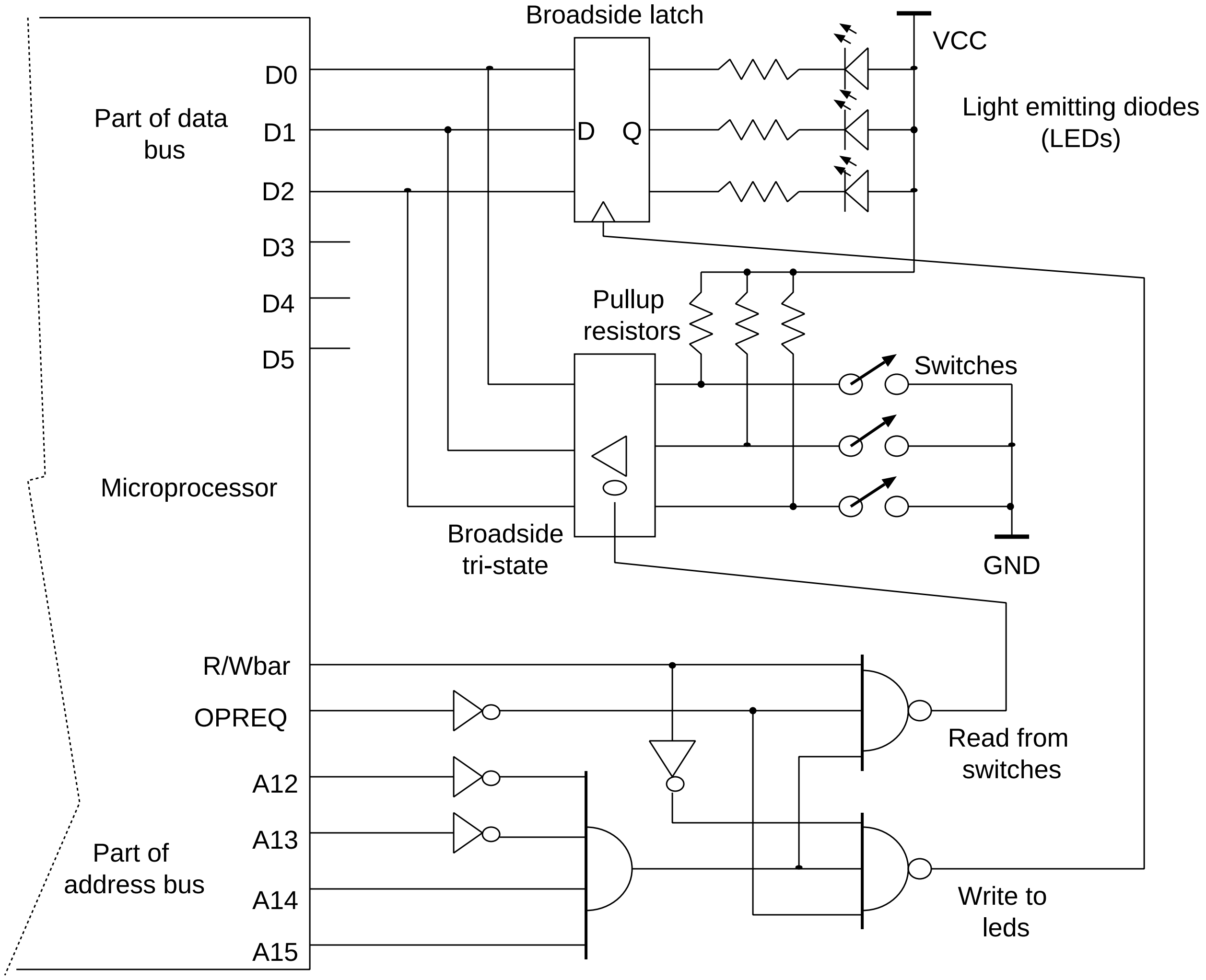
Logic Symbol



Internal Structure Block Diagram

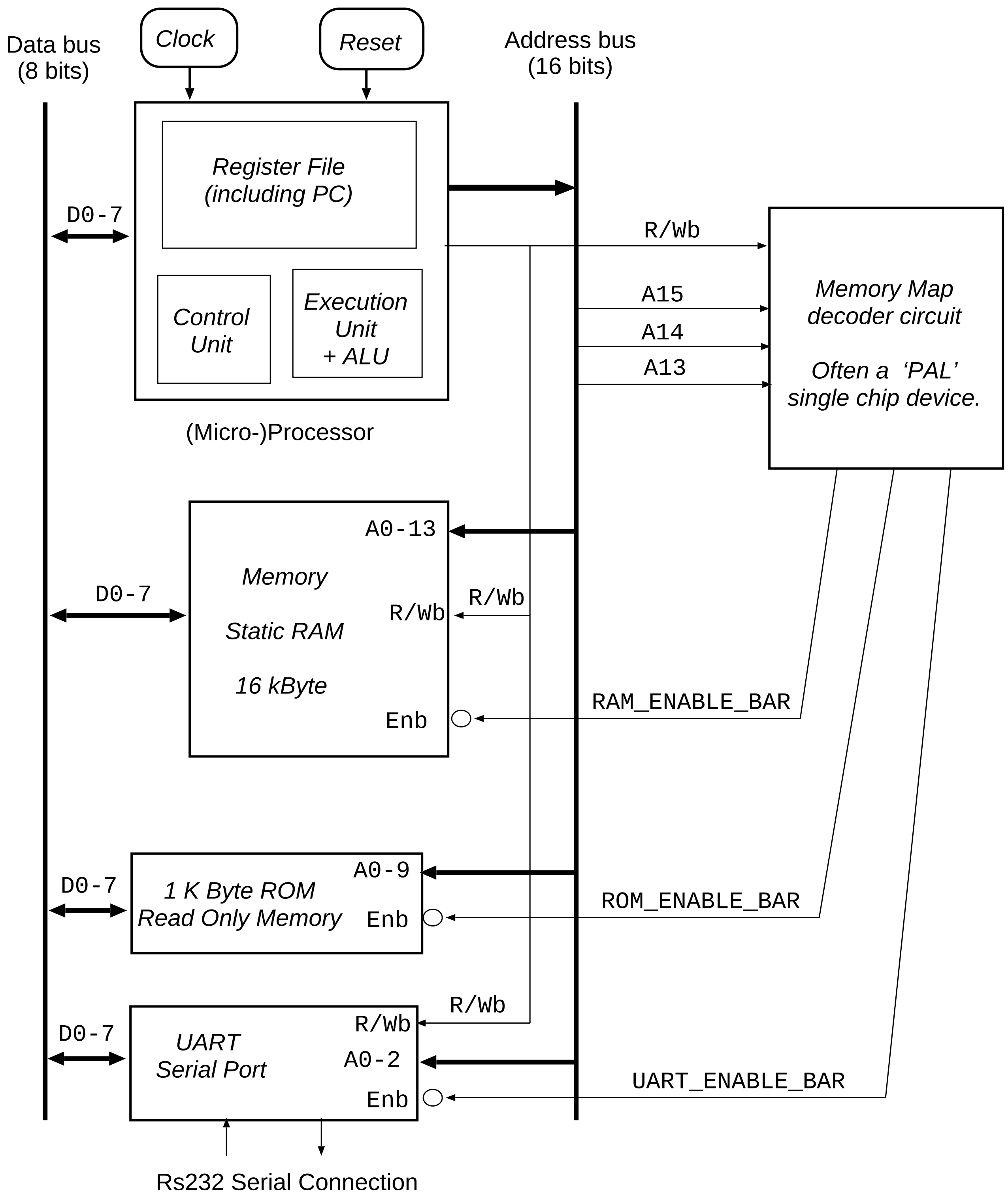
NB: Microprocessor internal details are not examinable for 1A.



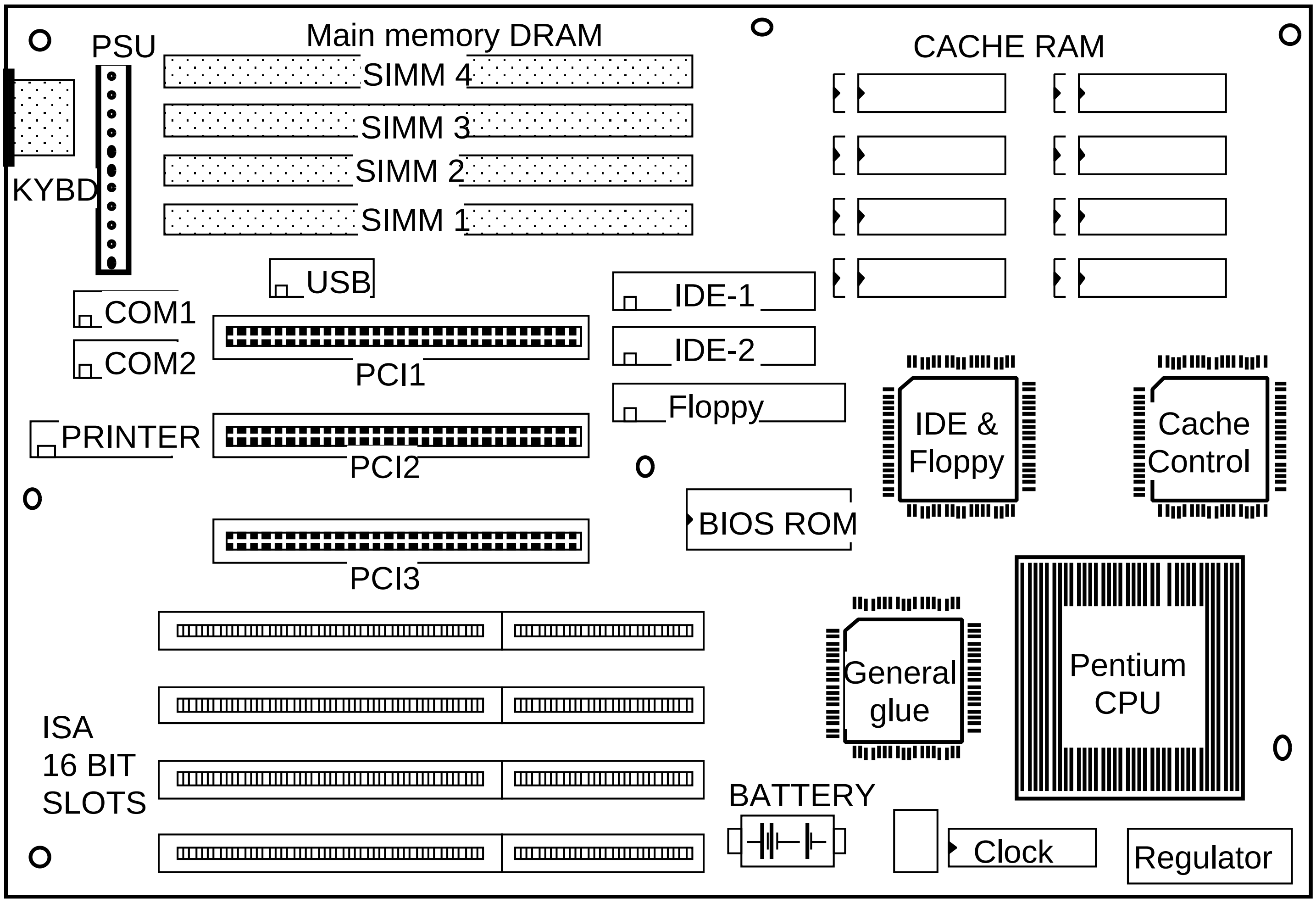


Example of memory address decode and simple LED and switch interfacing for programmed IO (PIO) to a microprocessor.

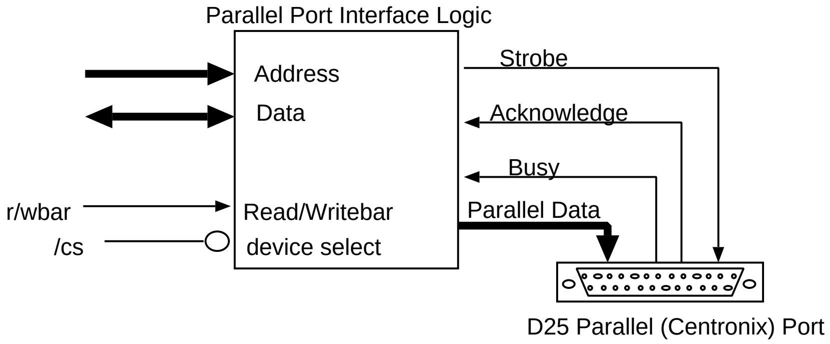
A small computer

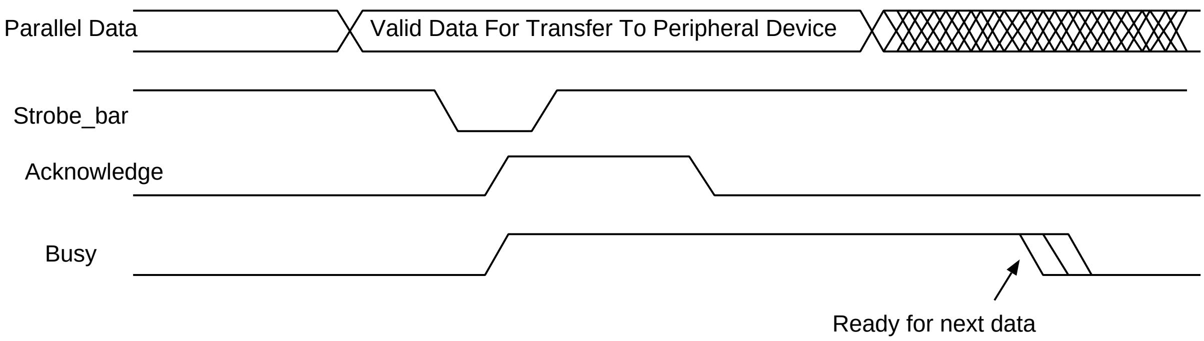


PC Motherboard, 1997 vintage

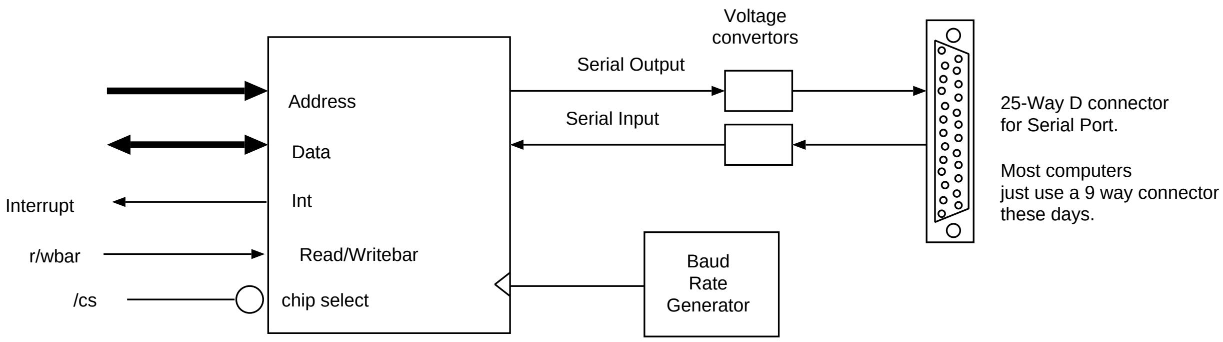


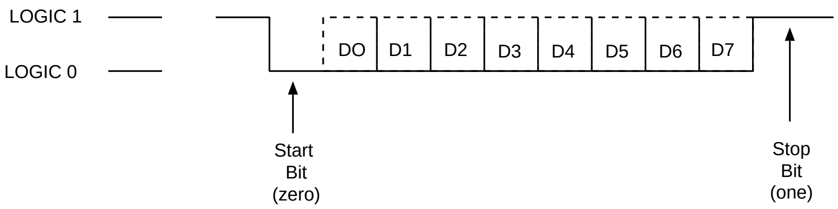
Parallel Port



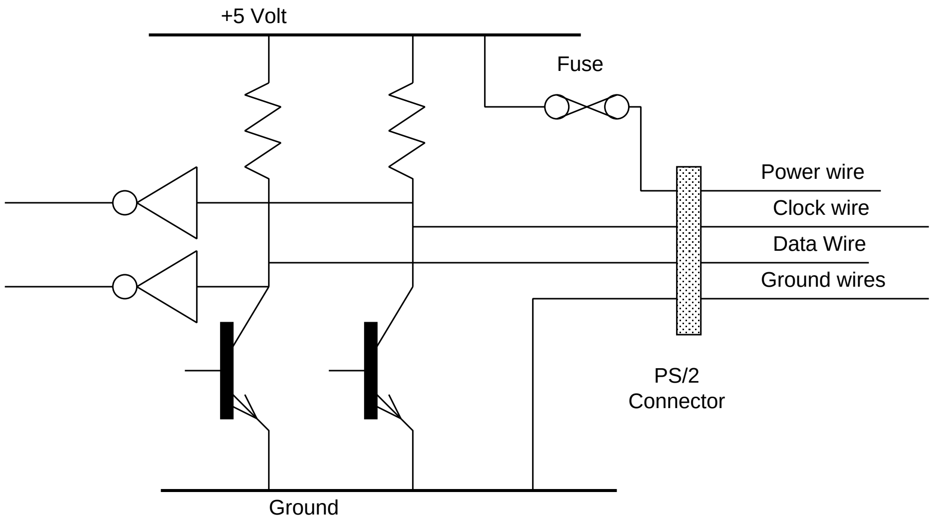


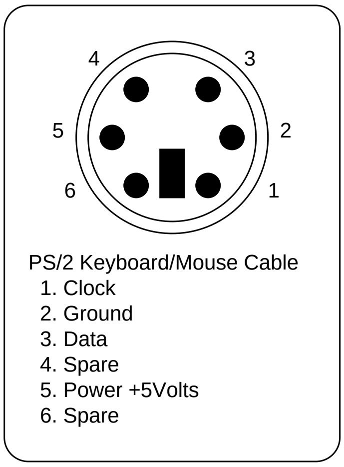
Serial Port (UART)


Voltage
convertors
Serial Output
Serial Input
Address
Data
Interrupt
Int
r/wbar
Read/Writebar
/cs
chip select
Baud
Rate
Generator
25-Way D connector
for Serial Port.
Most computers
just use a 9 way connector
these days.

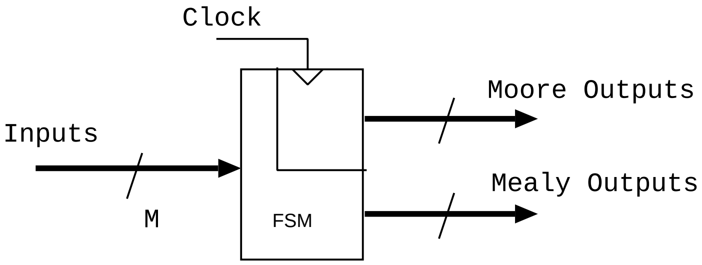

LOGIC 1
LOGIC 0
DO
D1
D2
D3
D4
D5
D6
D7
Start
Bit
(zero)
Stop
Bit
(one)

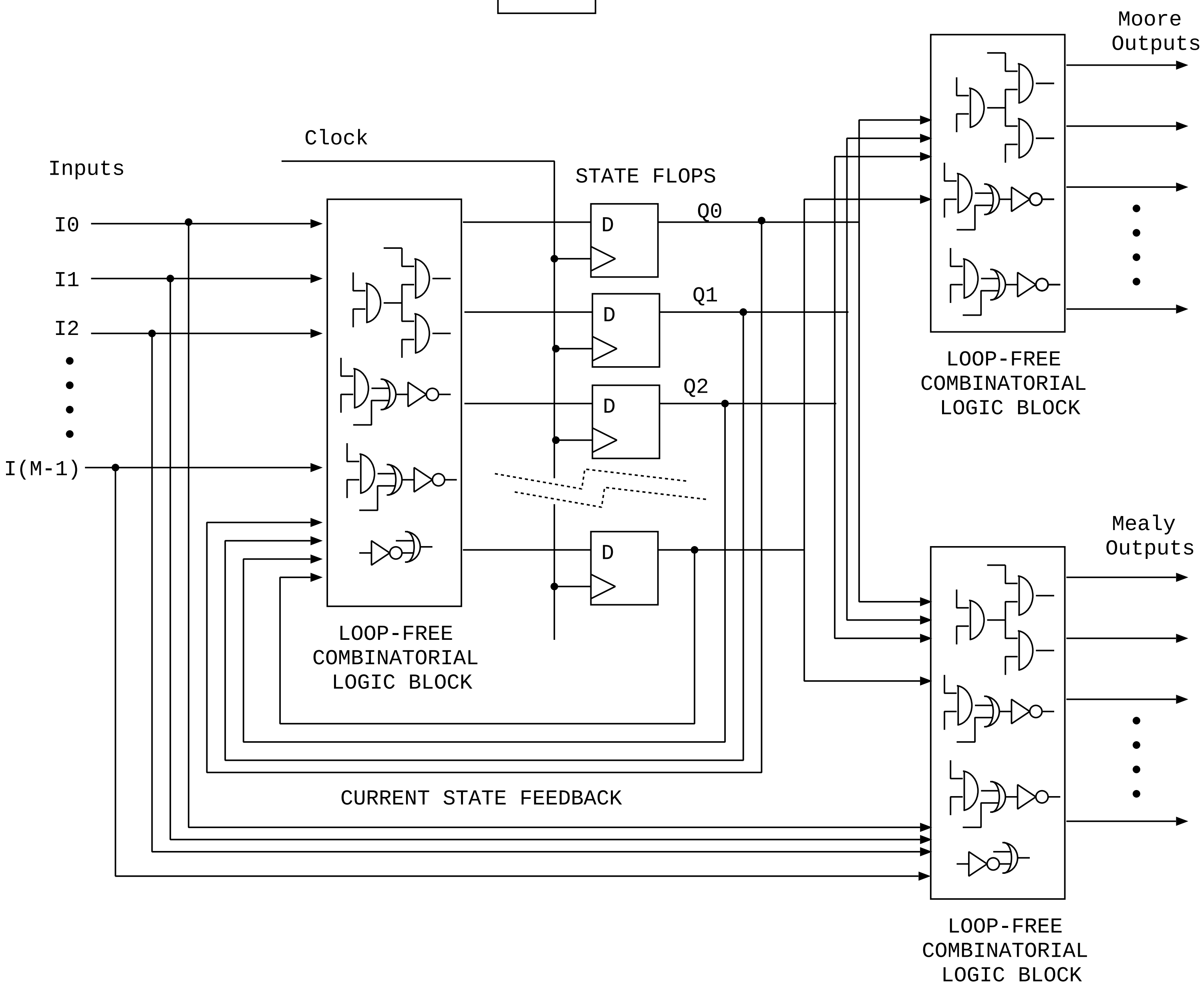
Keyboard and/or PS/2 port



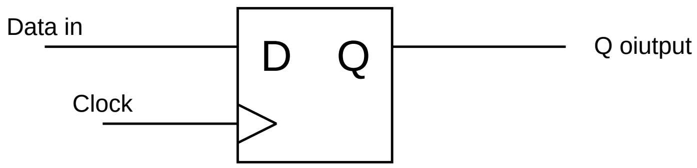


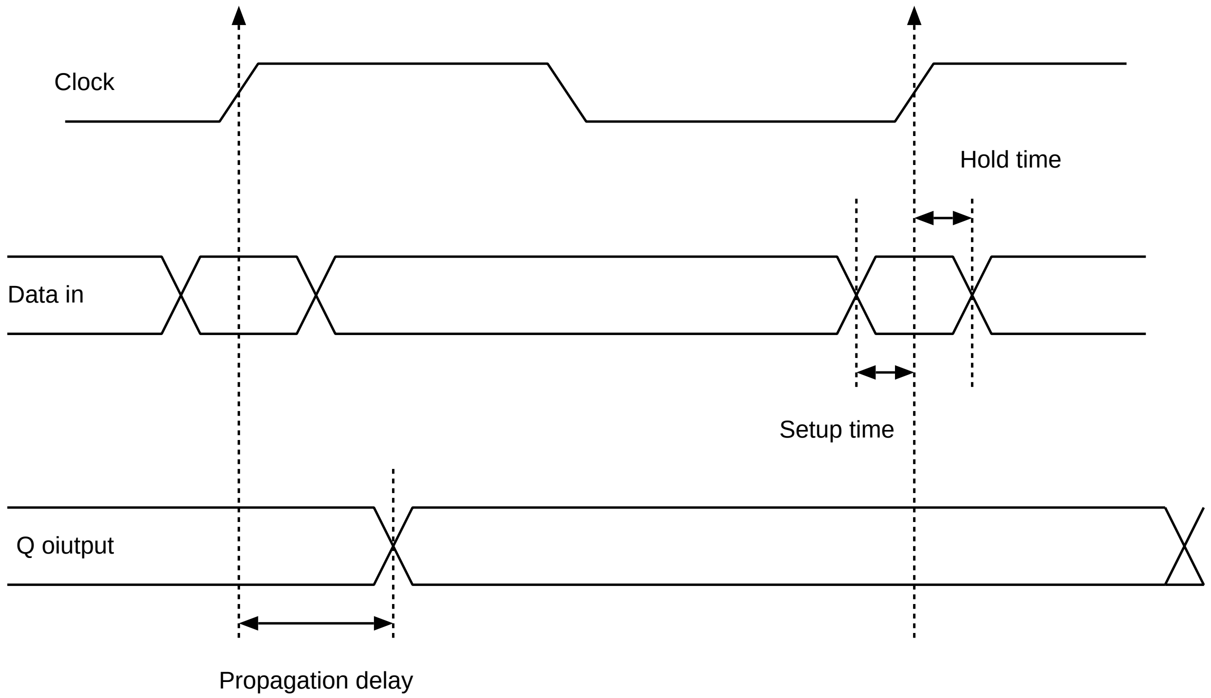
Canonical synchronous FSM



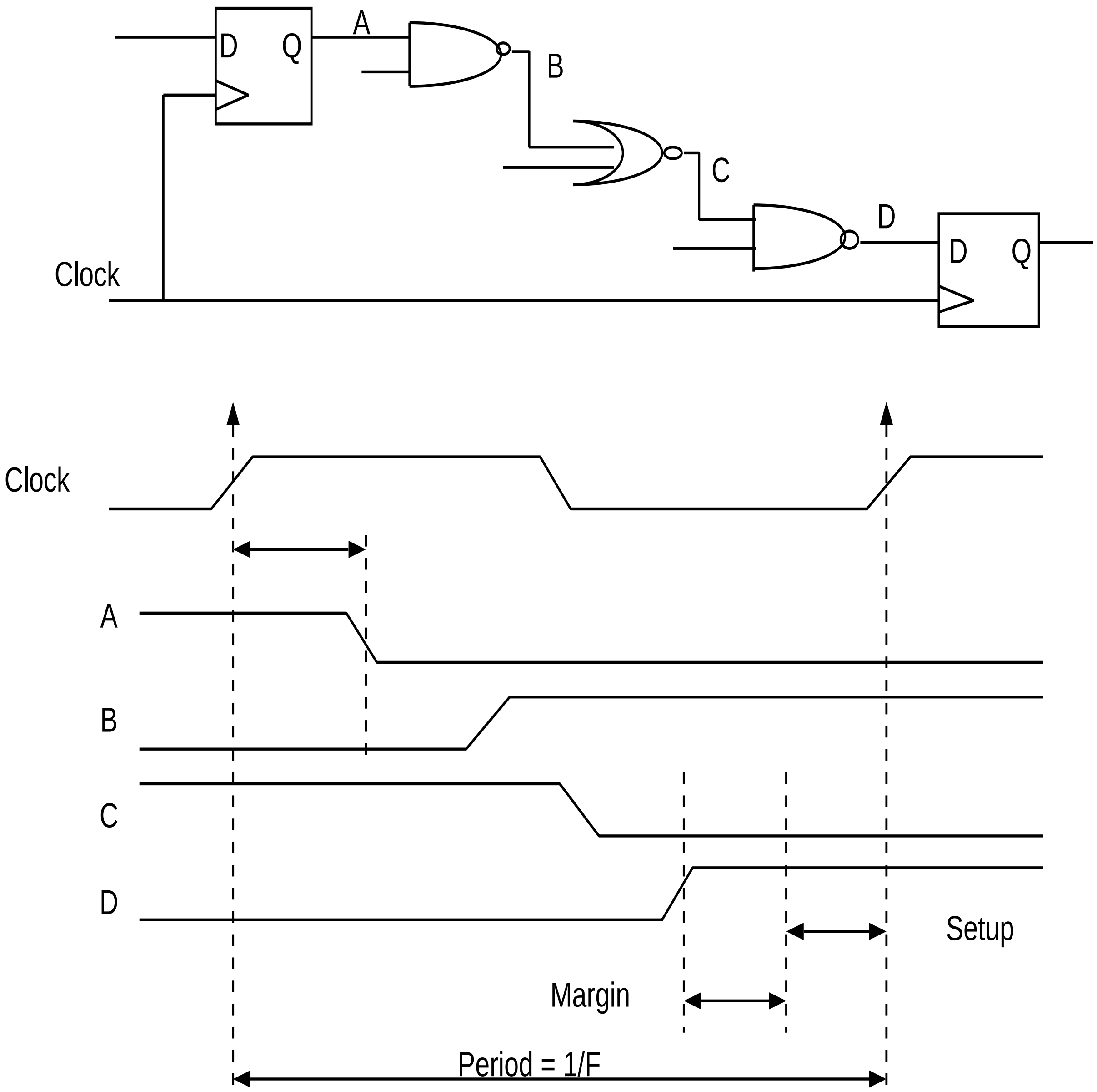


Timing Specifications

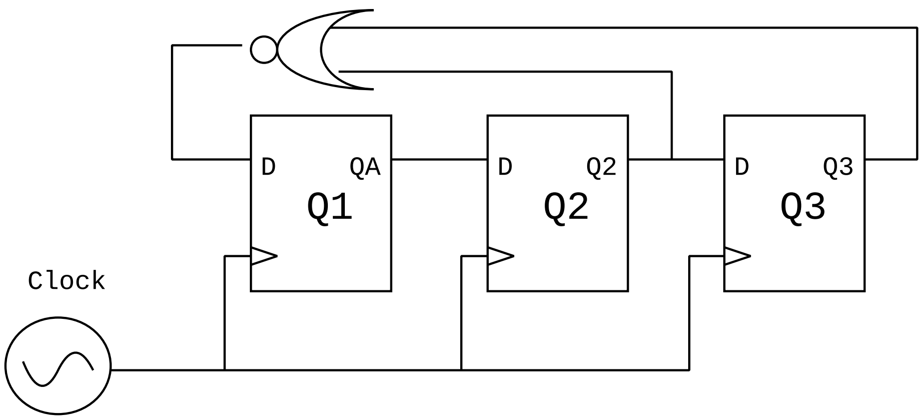




Typical nature of a critical path

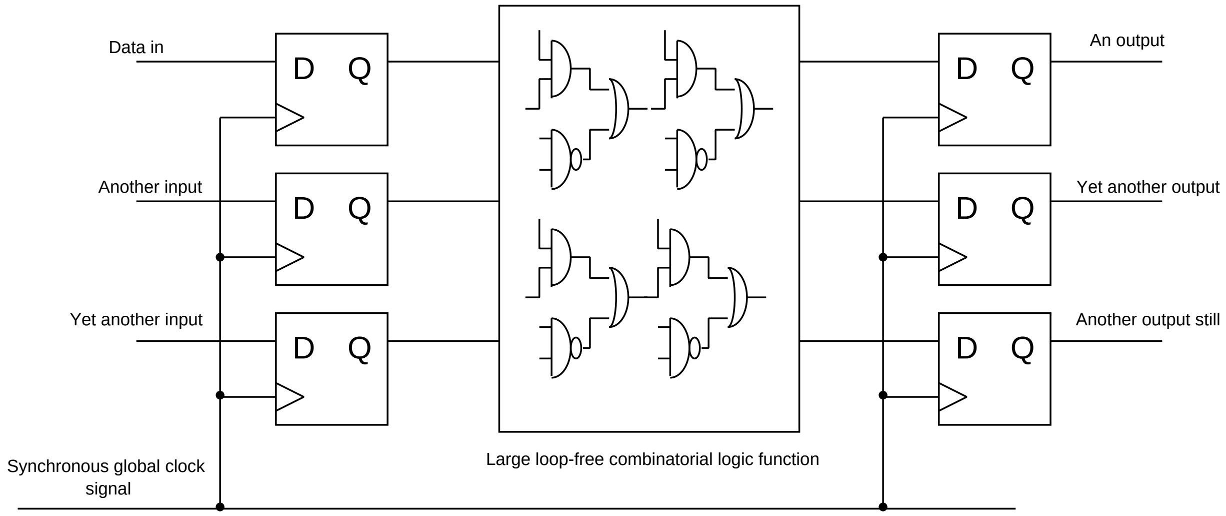


Johnson counters

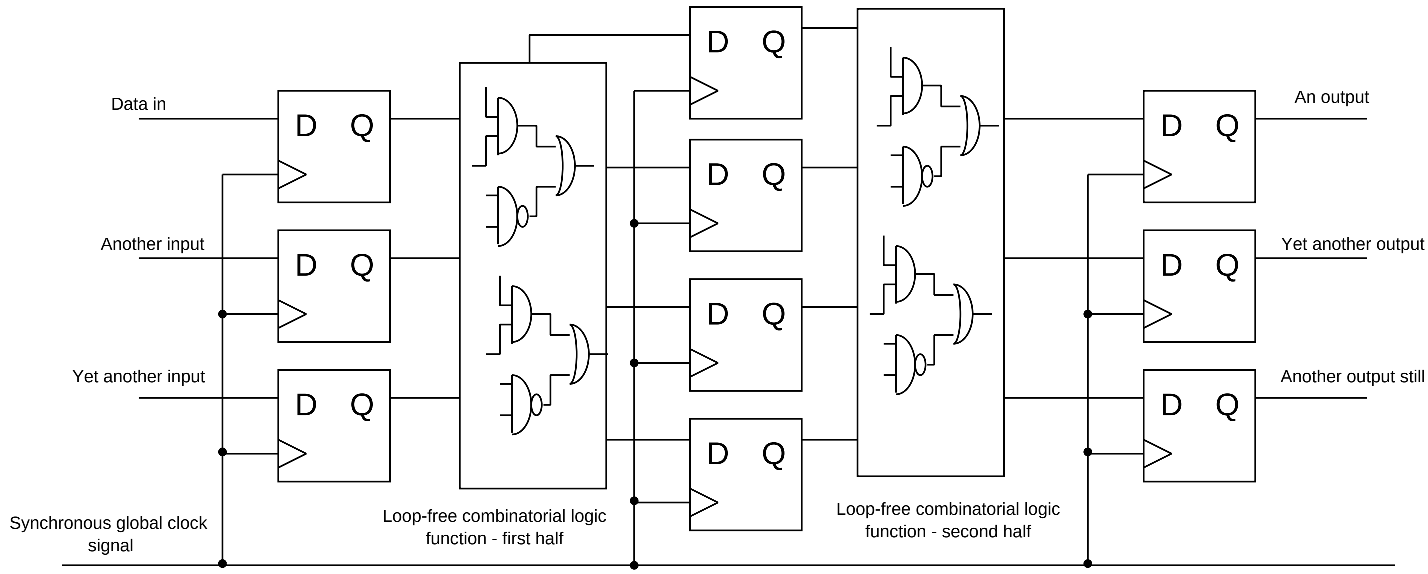


Q1	Q2	Q3

Pipelining

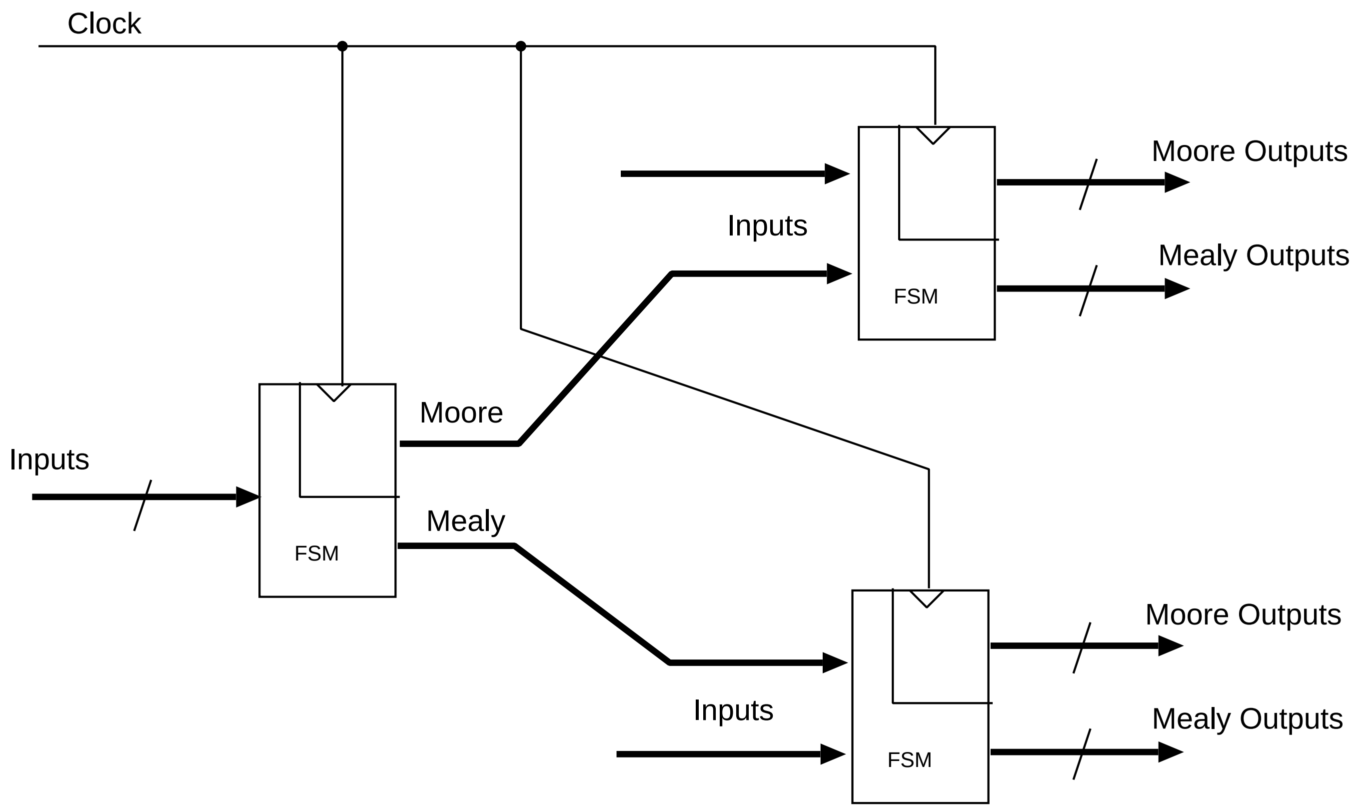


Desired logic function

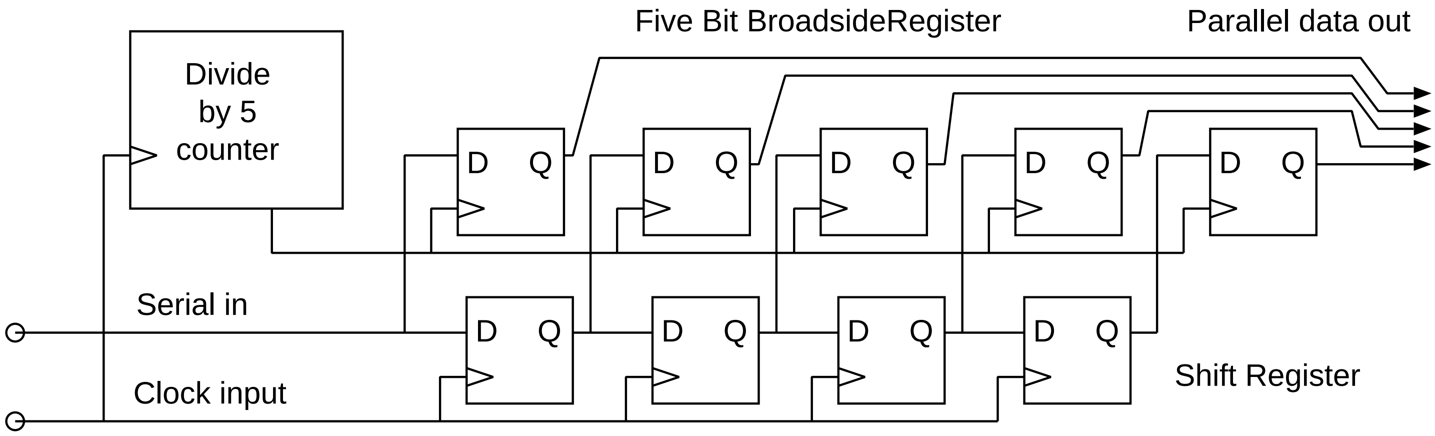


Desired logic function - pipelined version.

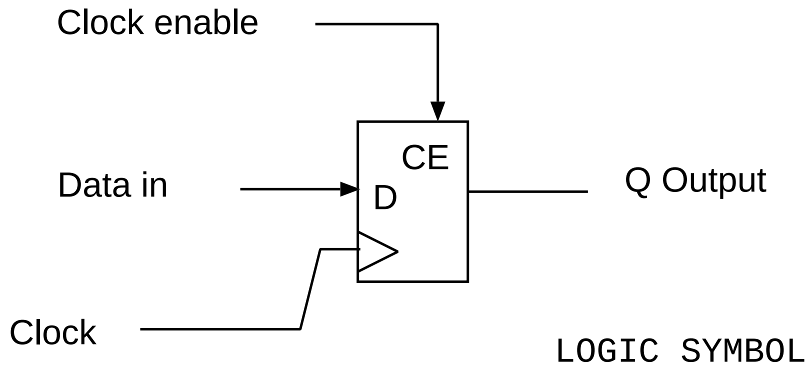
Cascading FSMs

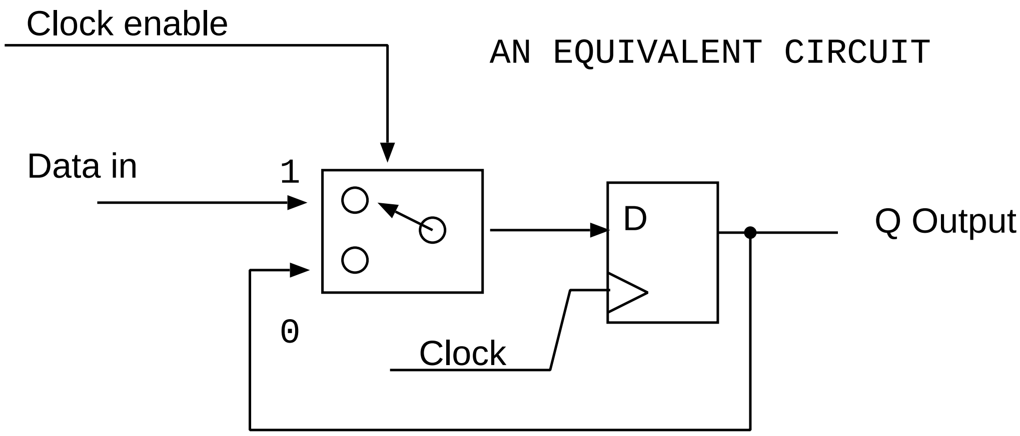


An example that uses (badly) a derived clock:
a serial-to-parallel converter

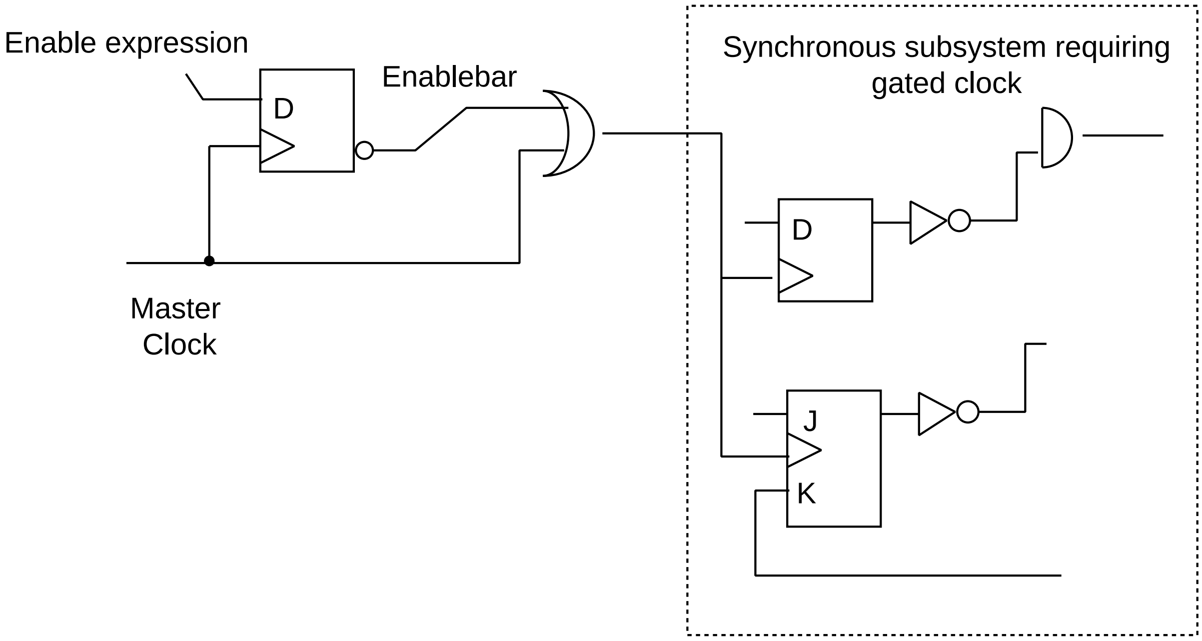


A D-type with clock-enable

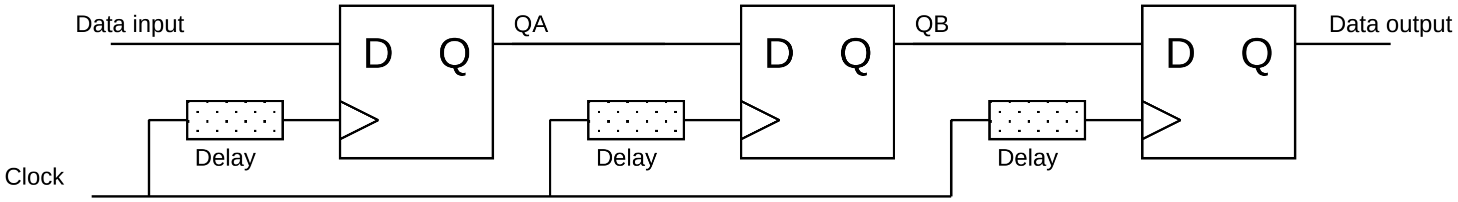




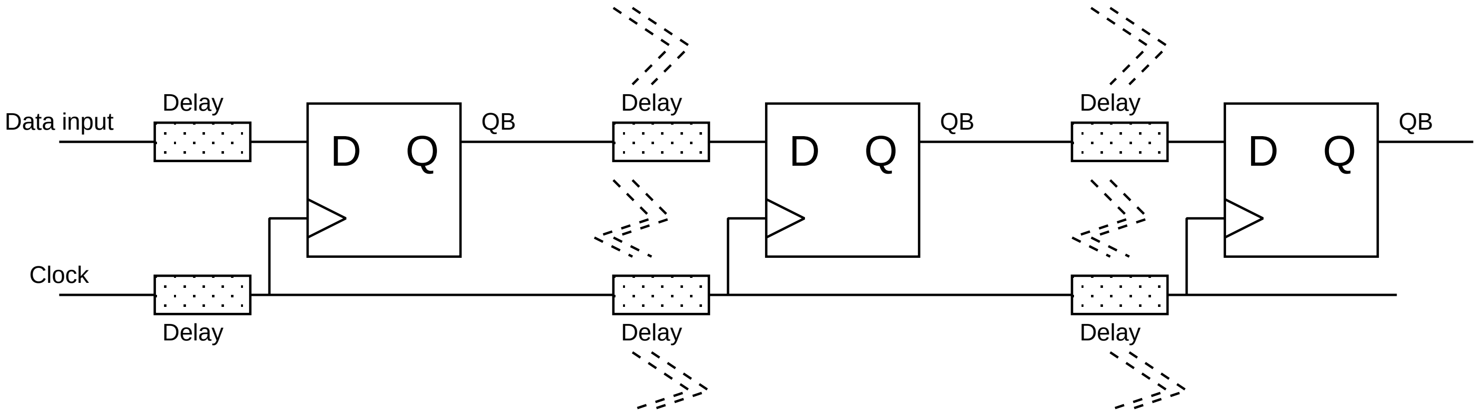
A Gated Clock



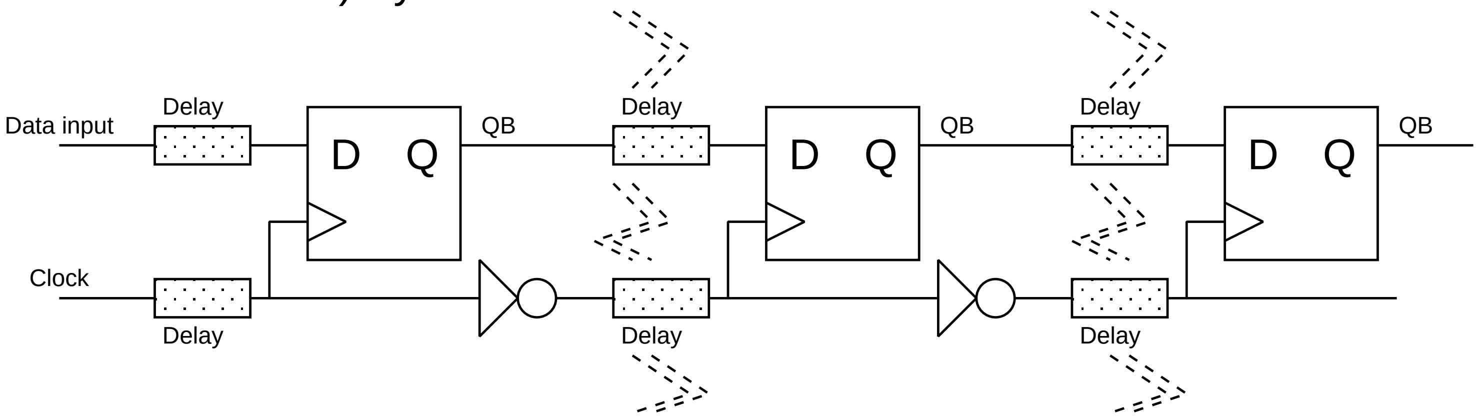
Clock Skew



a) A three-stage shift register with some clock skew delays.

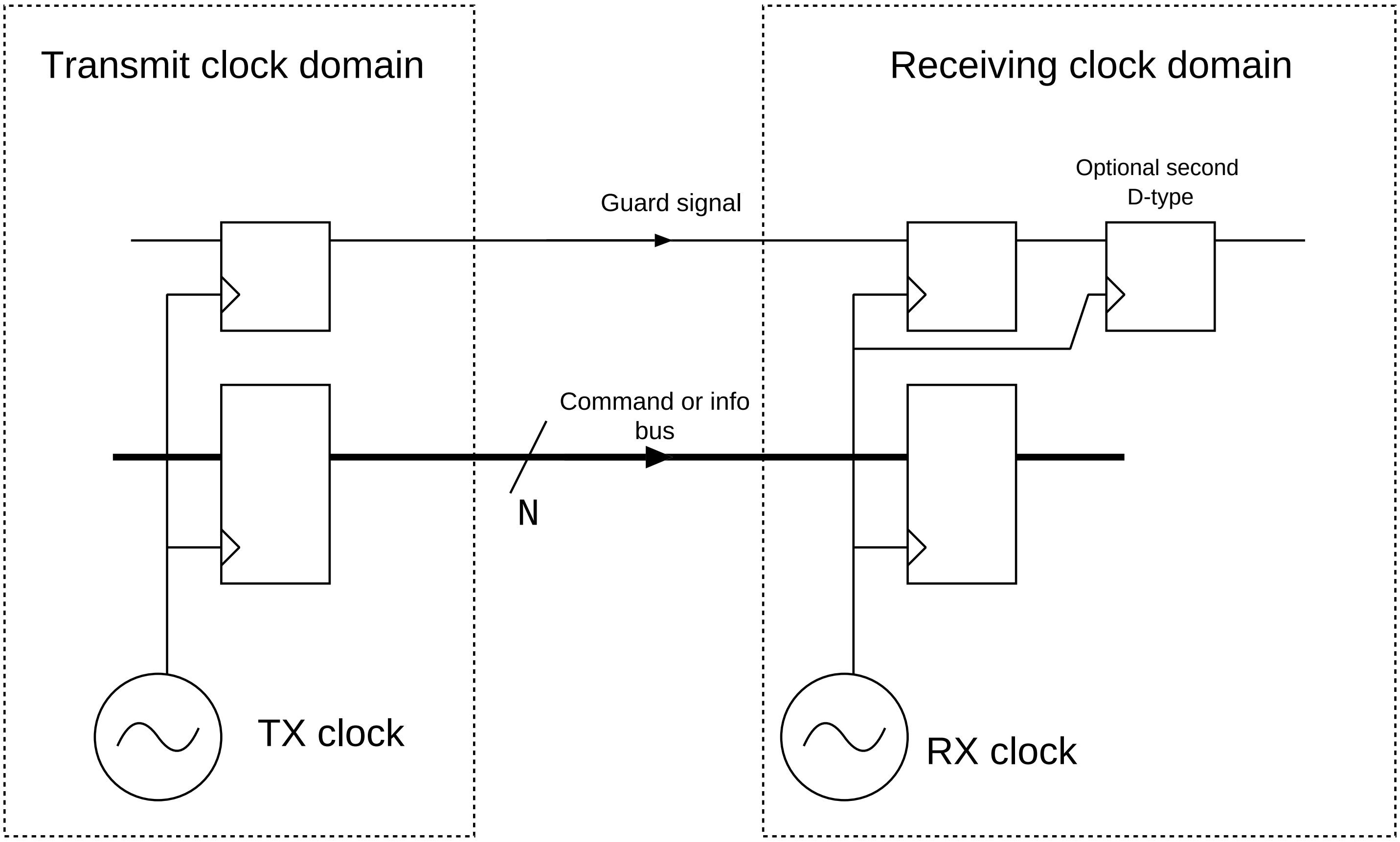


b) System interconnection with clock skews

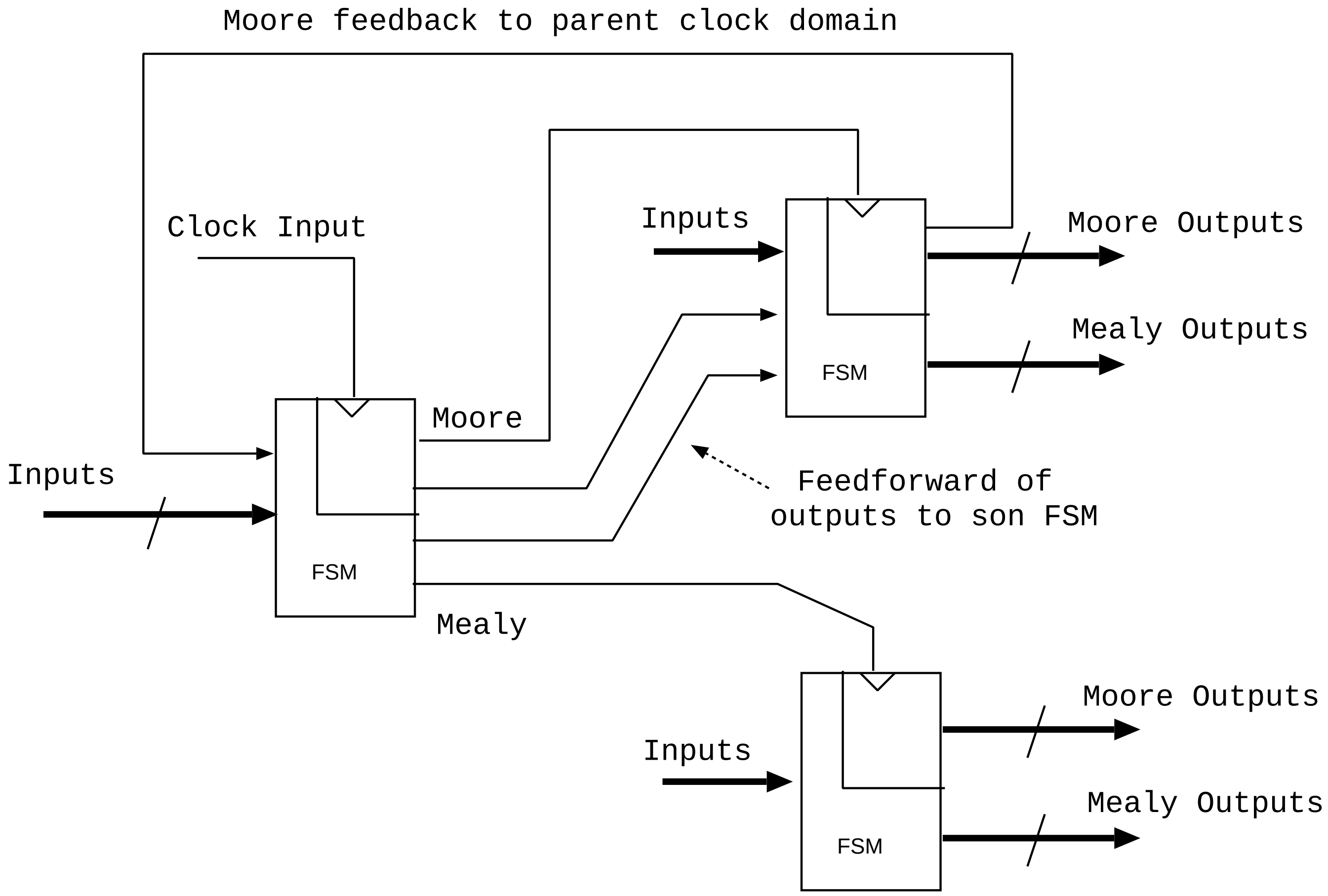


c) A solution for serious skew and delay problems ?

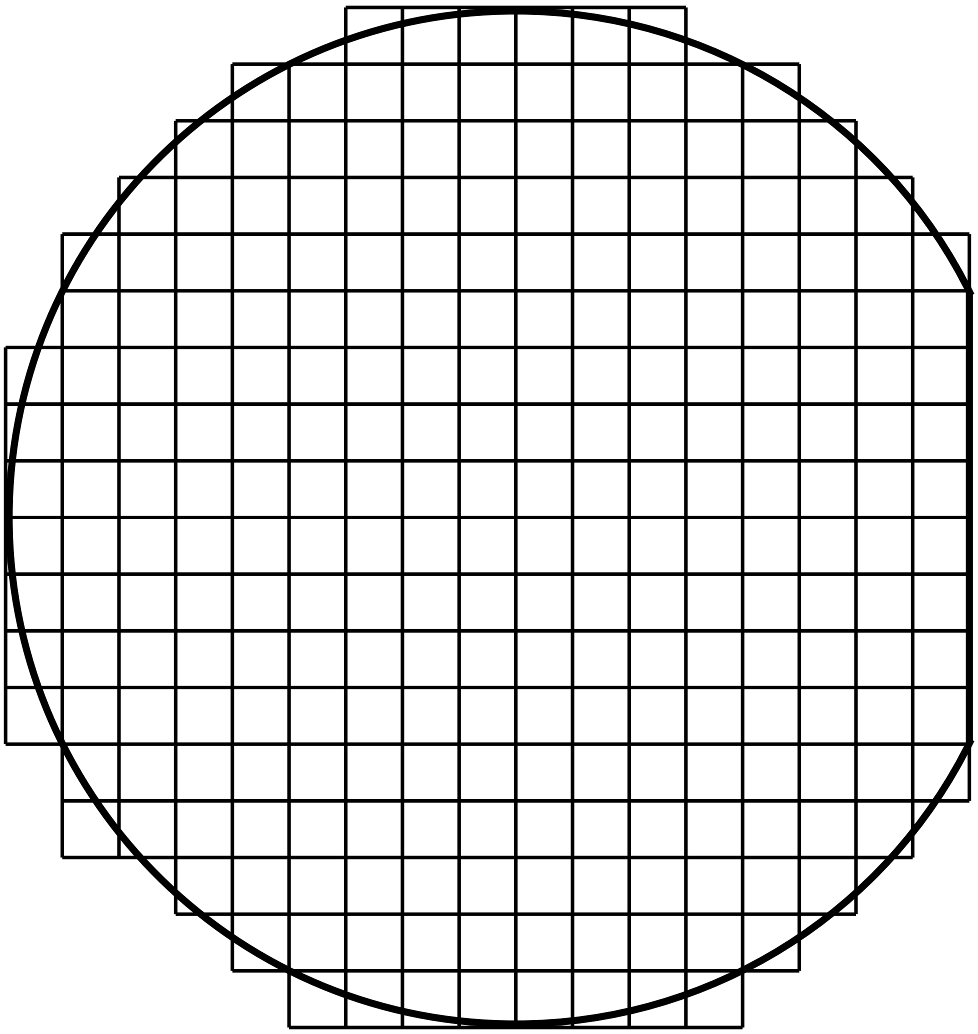
Crossing an async boundary



Paths between FSMs w/ derived clocks

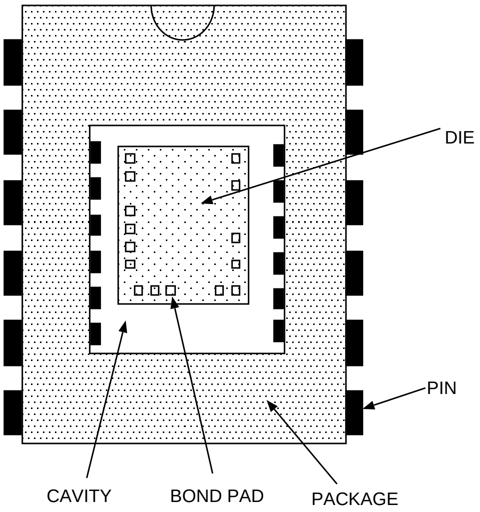


Dicing a wafer

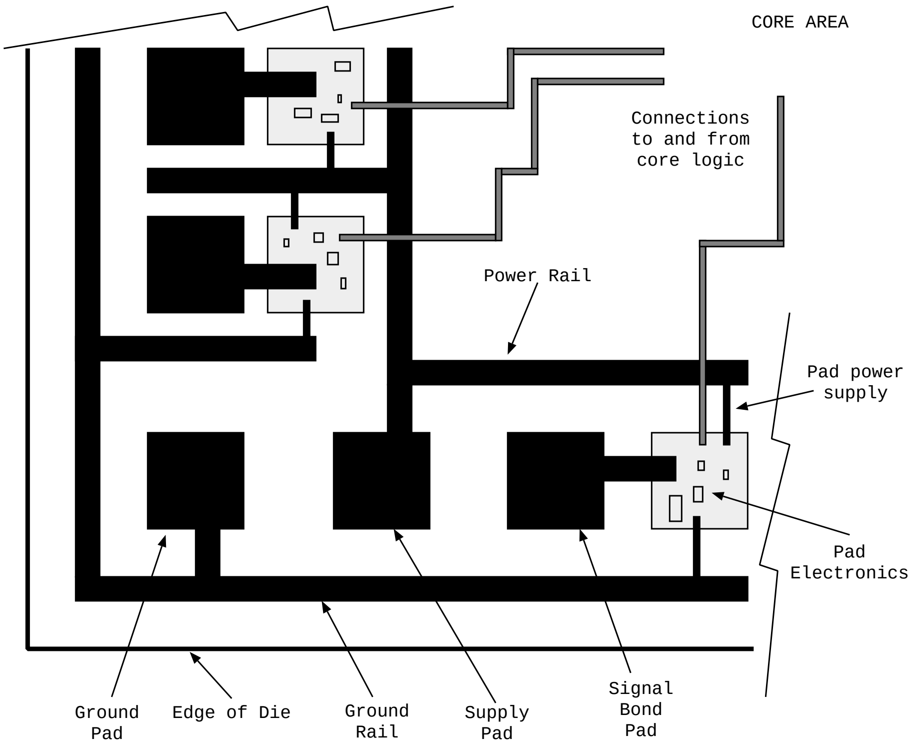


(Chips are not always square)

A chip in its package, ready for bond wires



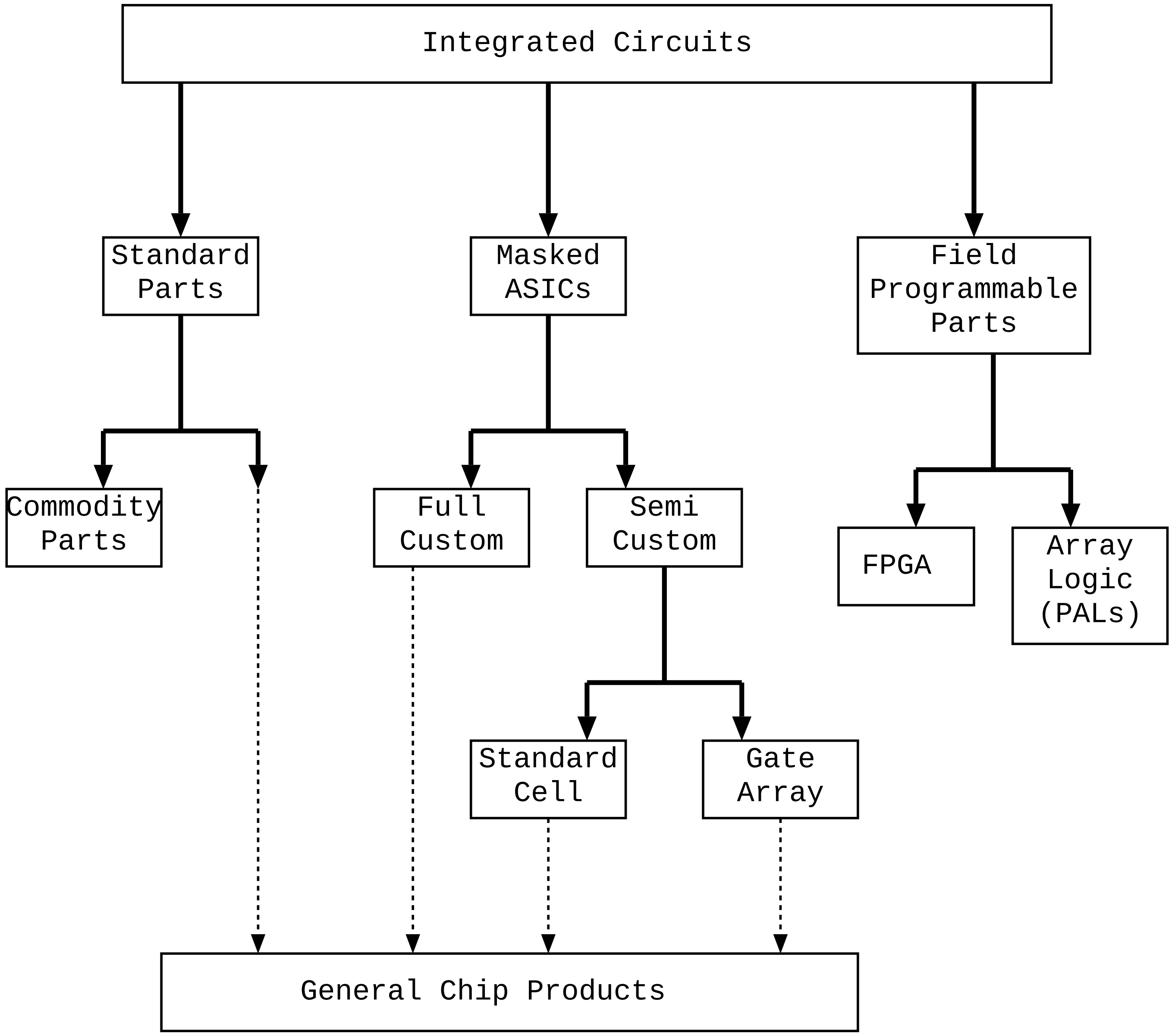
IO and power pads



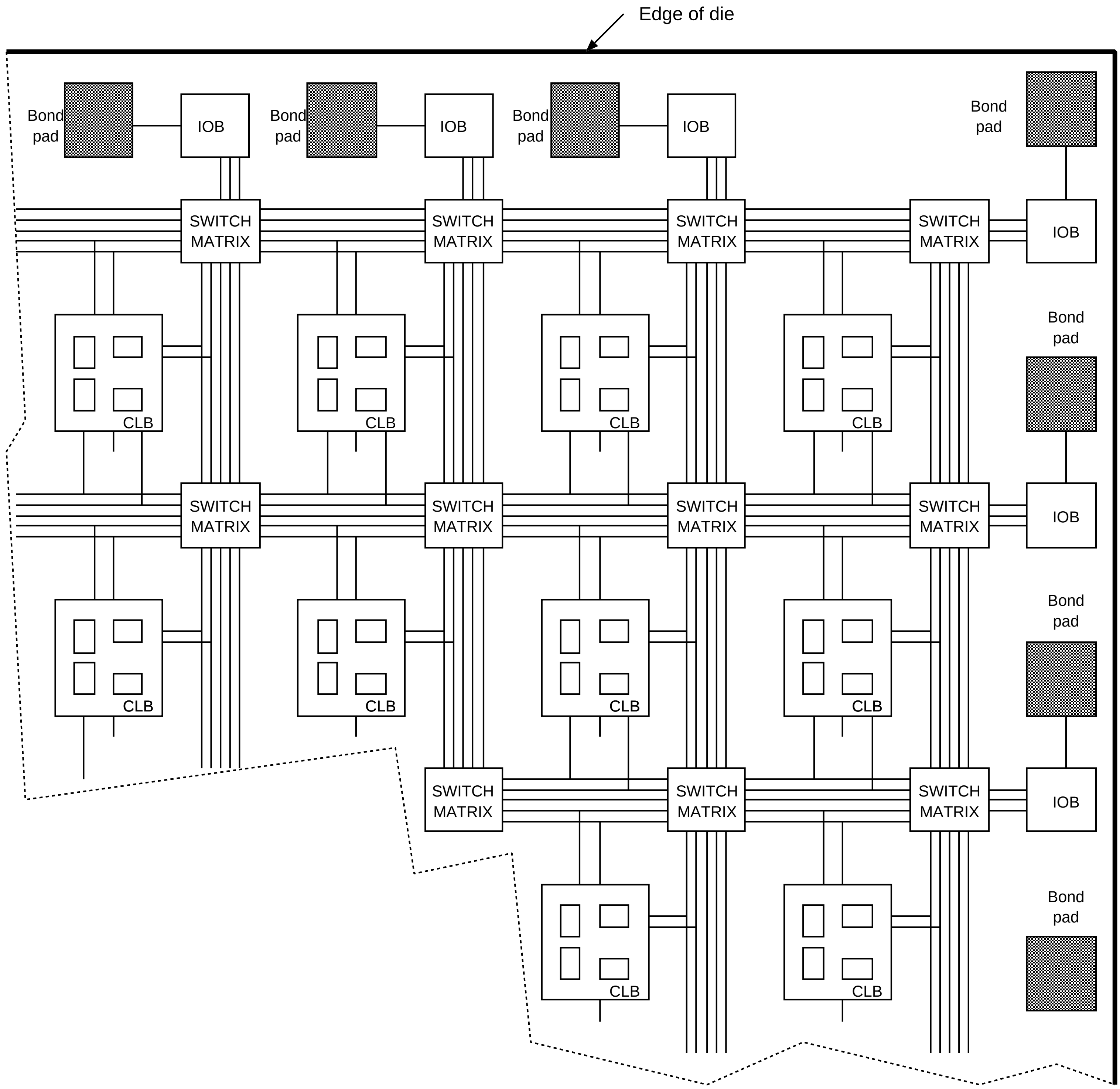
Die cost example

Area	Wafer dies	Working dies	Cost per working die
2	9000	8910	0.56
3	6000	5910	0.85
4	4500	4411	1.13
6	3000	2911	1.72
9	2000	1912	2.62
13	1385	1297	3.85
19	947	861	5.81
28	643	559	8.95
42	429	347	14.40
63	286	208	24.00
94	191	120	41.83
141	128	63	79.41
211	85	30	168.78
316	57	12	427.85
474	38	4	1416.89

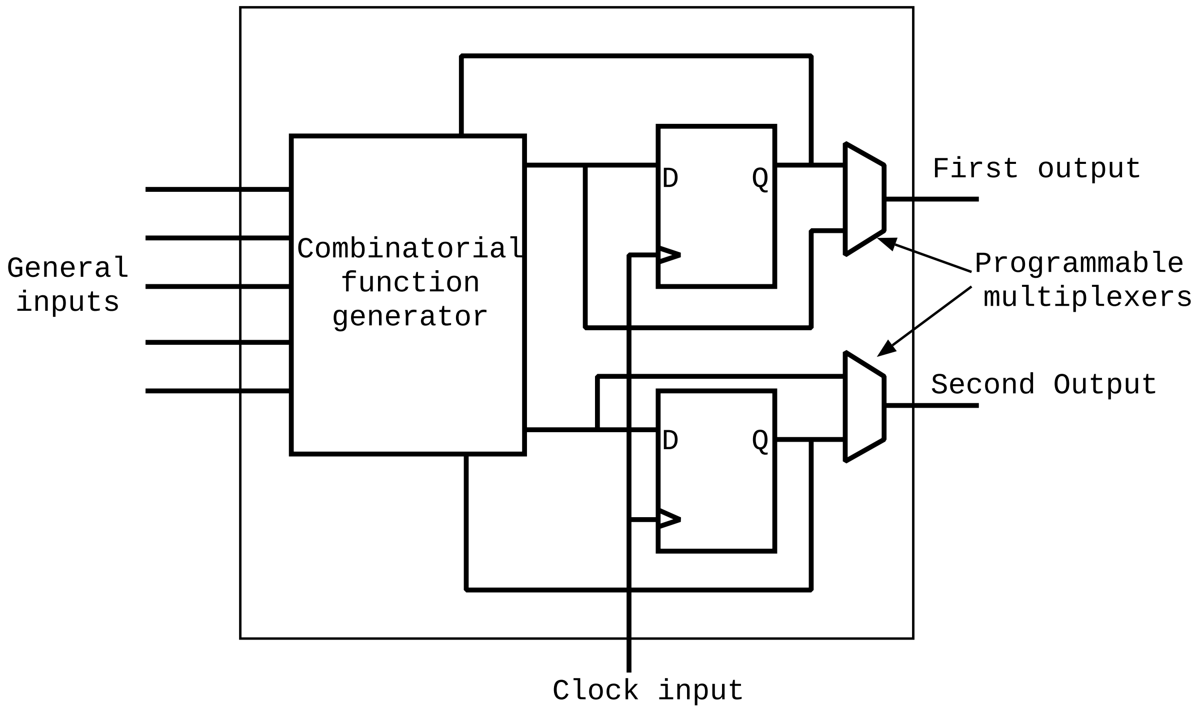
A taxonomy of ICs



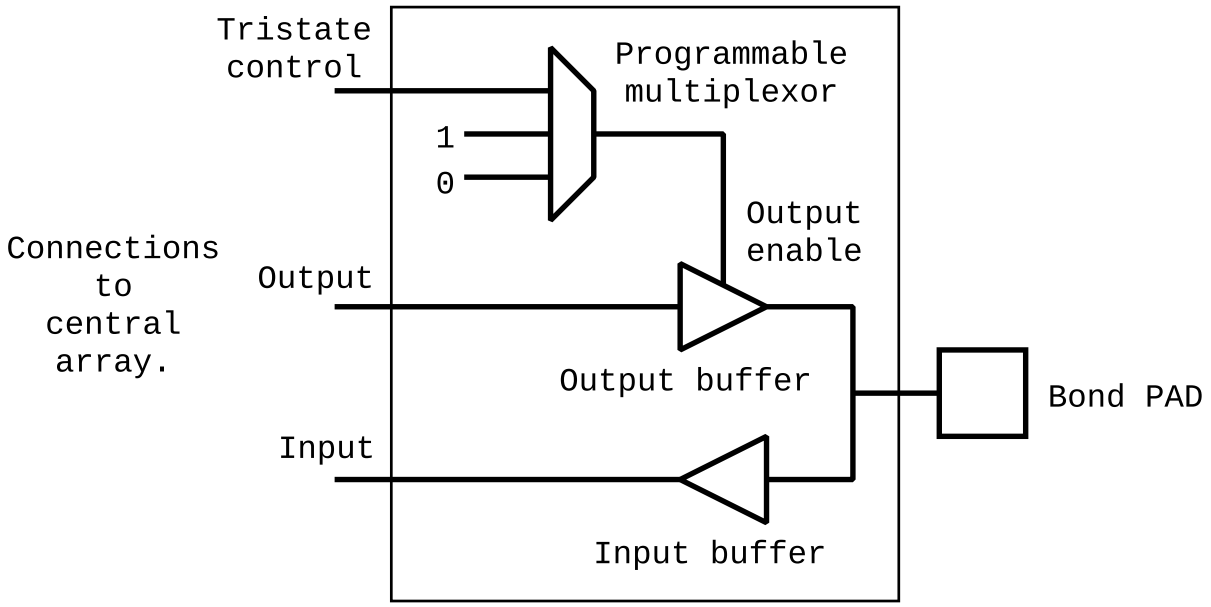
Field Programmable Gate Arrays

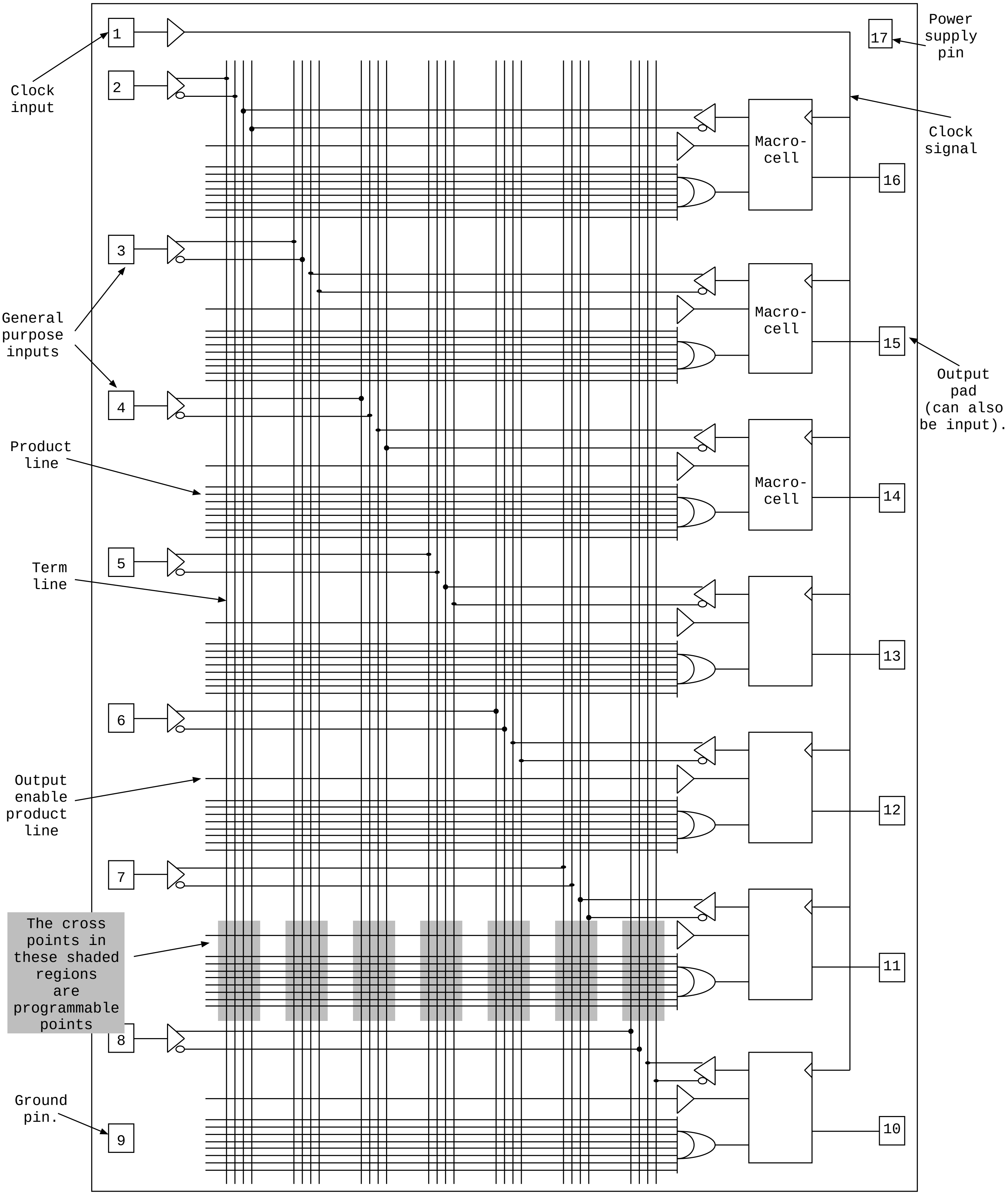


A configurable logic block for a look-up-table based FPGA



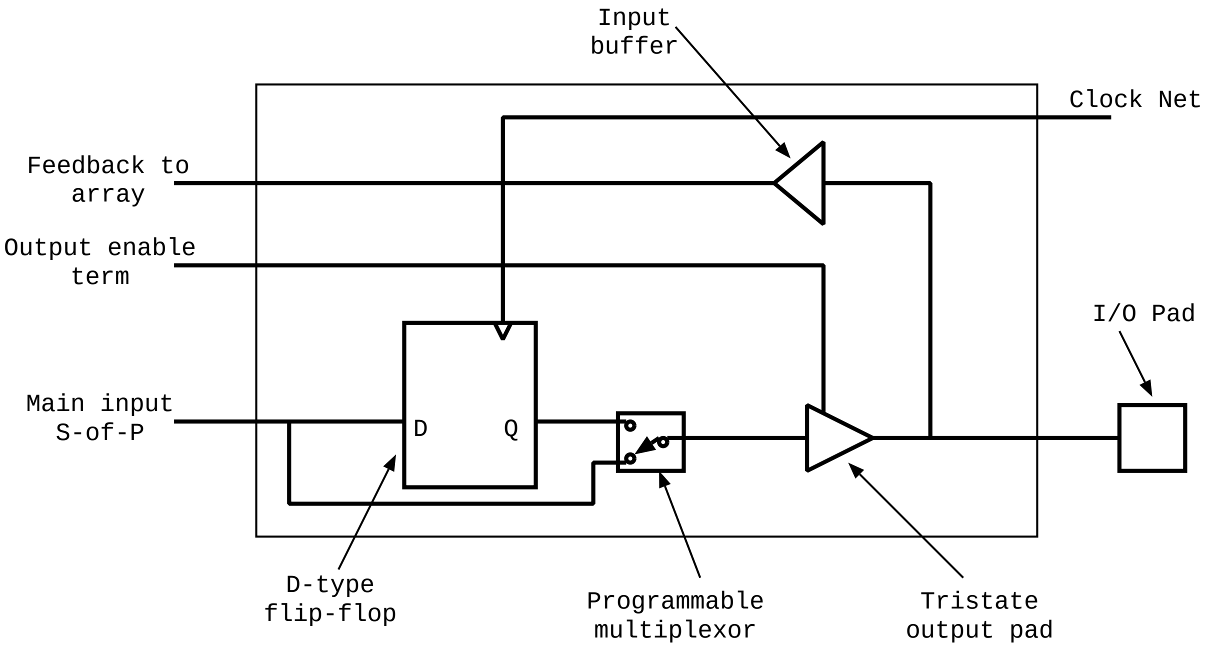
A simple IO block FPGA





Clock input
General purpose inputs
Product line
Term line
Output enable product line
The cross points in these shaded regions are programmable points
Ground pin.
Power supply pin
Clock signal
Output pad (can also be input).
Macro-cell
Macro-cell
Macro-cell
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17

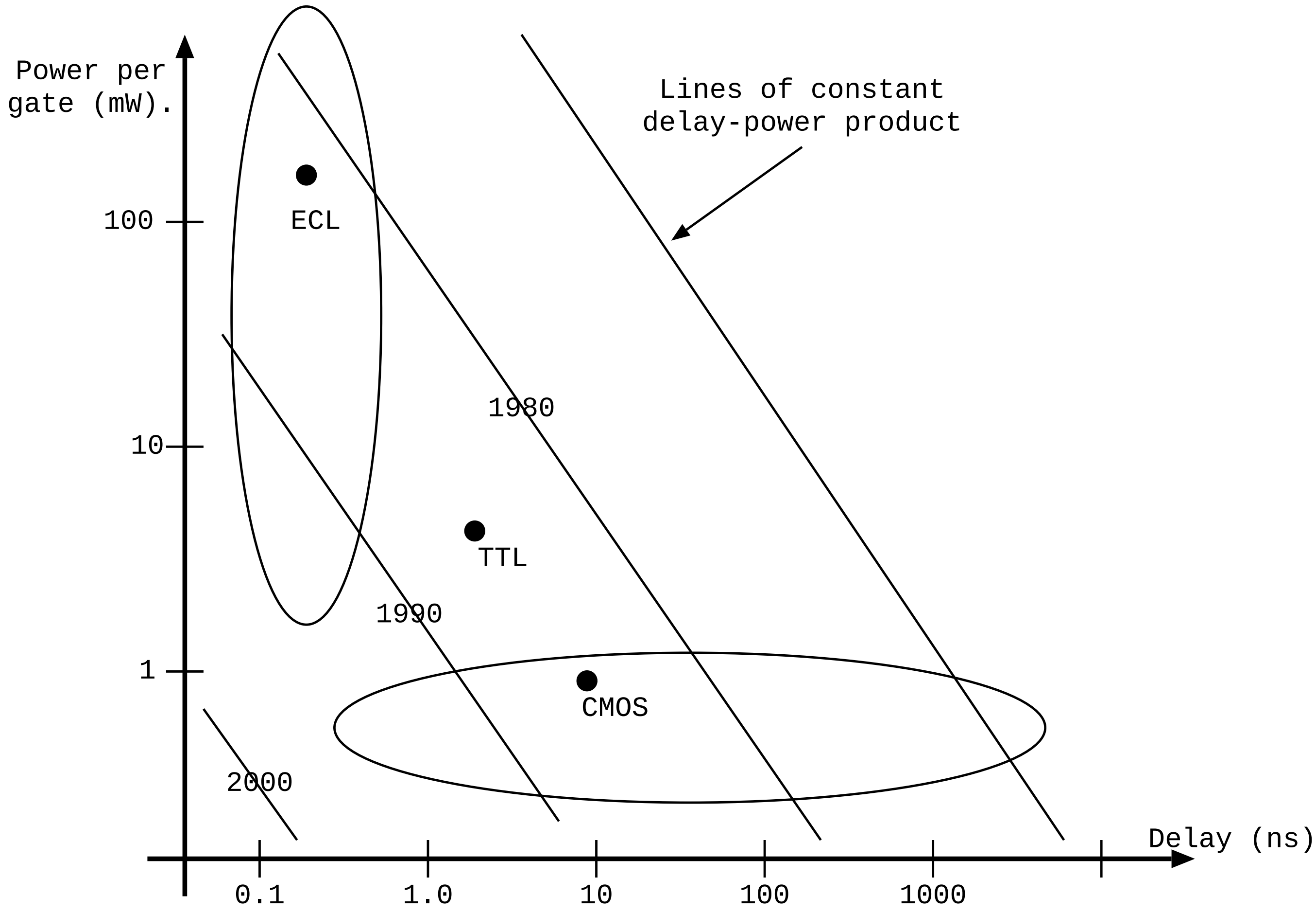
Contents of the PAL macrocell



Example programming of a PAL showing only fuses for the top macrocell

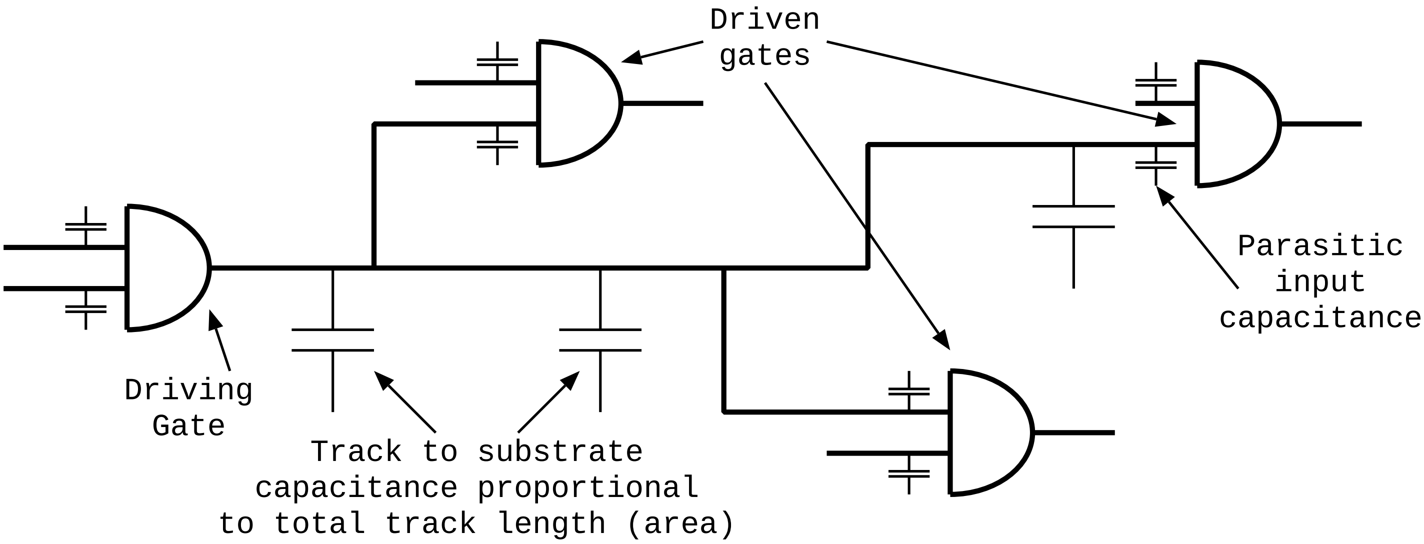
```
pin 16 = o1;
pin 2 = a;
pin 3 = b;
pin 4 = c

o1.oe = ~a;
o1 = (b & o1) | c;

-x-- ---- ---- ---- ---- ---- ----  (oe term)
--x- x--- ---- ---- ---- ---- ----  (pin 3 and 16)
---- ---- x--- ---- ---- ---- ----  (pin 4)
xxxx xxxx xxxx xxxx xxxx xxxx xxxx
xxxx xxxx xxxx xxxx xxxx xxxx xxxx
xxxx xxxx xxxx xxxx xxxx xxxx xxxx
xxxx xxxx xxxx xxxx xxxx xxxx xxxx
xxxx xxxx xxxx xxxx xxxx xxxx xxxx
x                                   (macrocell fuse)
```

Delay-power style of technology comparison chart



technology	device	propagation delay (ns)	power (mW)	product (pJ)
CMOS	74hc00	7 ns	1 mW	7
TTL	74f00	3.4 ns	5 mW	17
ECL	sp92701	0.8 ns	200 mW	160

Logic net with tracking and input load capacitances



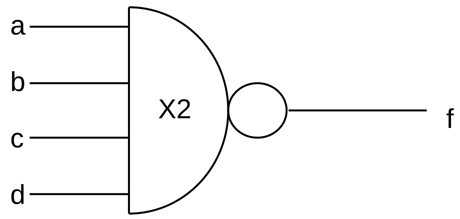
An example cell from a manufacturer's cell library

NAND4 Standard Cell

Library: CBG0.5um

4 input NAND gate with x2 drive

Schematic Symbol



Simulator/HDL Call

NAND4X2(f, a, b, c, d);

Logical Function

F = NOT(a & b & c & d)

ELECTRICAL SPECIFICATION

Switching characteristics : Nominal delays (25 deg C, 5 Volt, signal rise and fall 0.5 ns)

Inputs	Outputs	O/P Falling		O/P Rising	
		(ps)	ps/LU	ps	ps/LU
A	F	142	37	198	33
B	F	161	37	249	33
C	F	165	37	293	33
D	F	170	37	326	34

Min and Max delays depend upon temperature range, supply voltage, input edge speed and process spreads. The timing information is for guidance only. Accurate delays are used by the UDC.

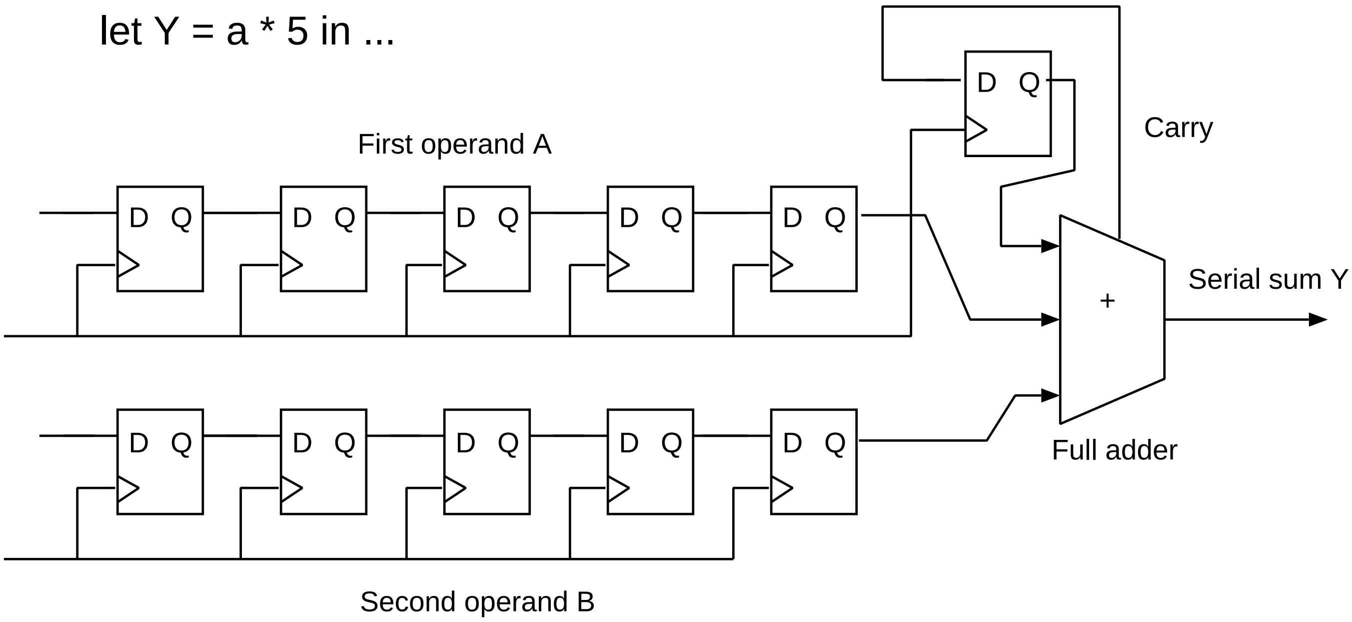
CELL PARAMETERS : (One load unit = 49 fF)

Parameters	Pin	Value	Units
Input loading	a b c d	2.1 2.1 2.1 2.0	Load units
Drive capability	f	35	Load units

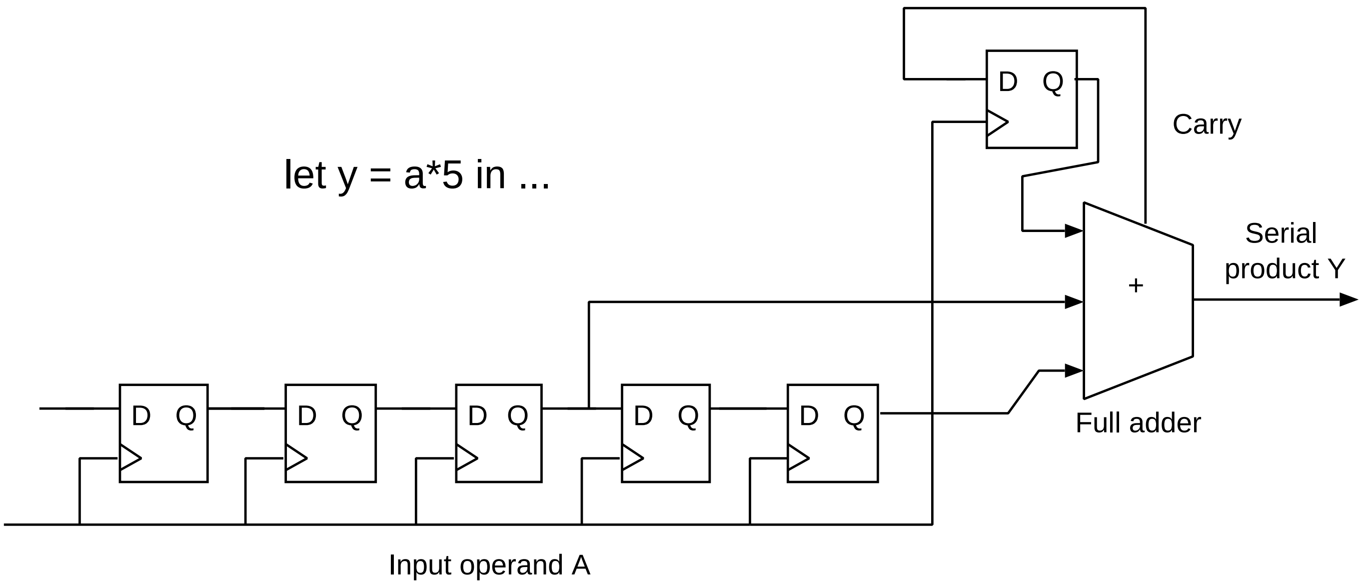
Comparative view of digital logic technologies

Technology	Maximum clock speed	Maximum gate count	Maximum I/Os
GaAs bipolar	100 GHz	500	30
GaAs fet	30 GHz	10K	300
Si ECL	10 GHz	10M	500
Si CMOS	8 GHz	50M	1000
Within Si CMOS			
Full-custom	8 GHz	50M	1000
Standard cell	4 GHz	25M	1000
Gate array	2 GHz	5M	1000
FPGA	175 MHz	500K	800
CPLD	150 MHz	10K	200
PAL	200 MHz	500	60

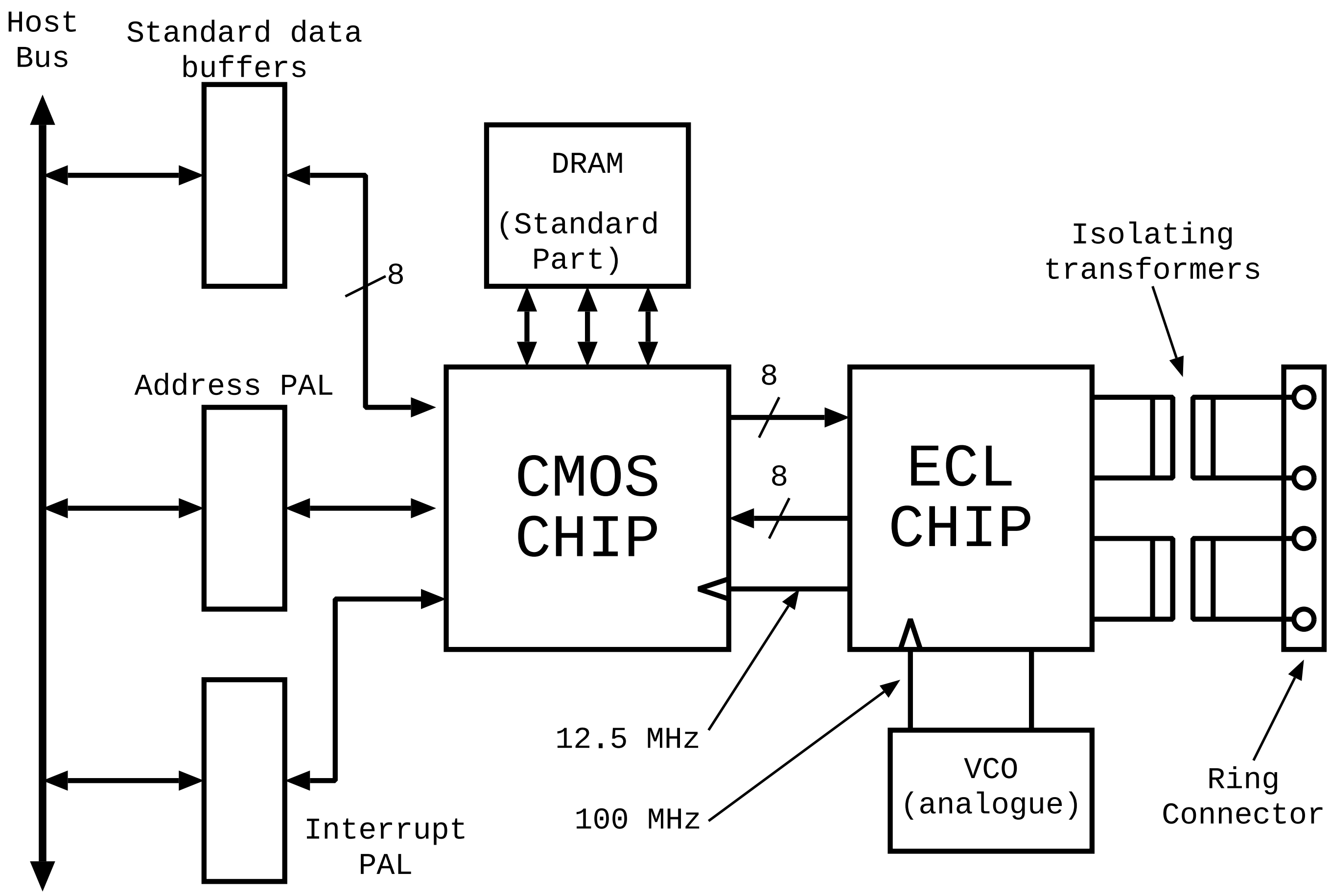
Addition of two integers serially, l.s.b first



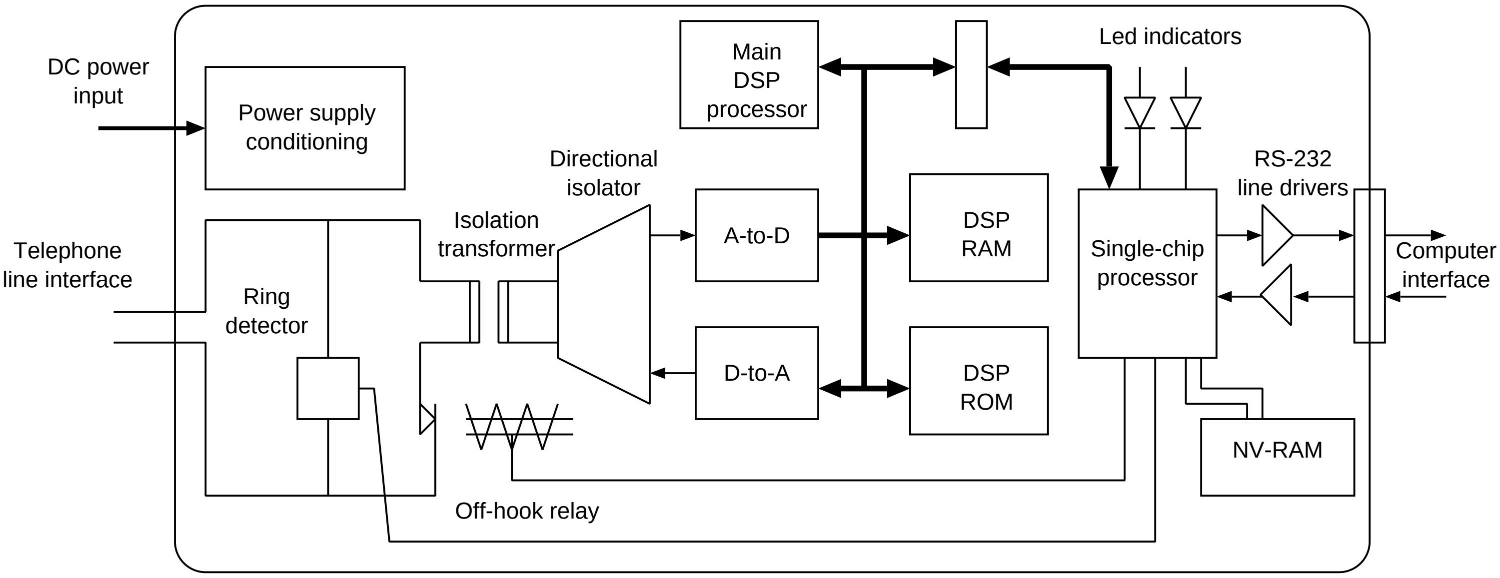
Bit-serial multiplication of an integer by a hardwired constant



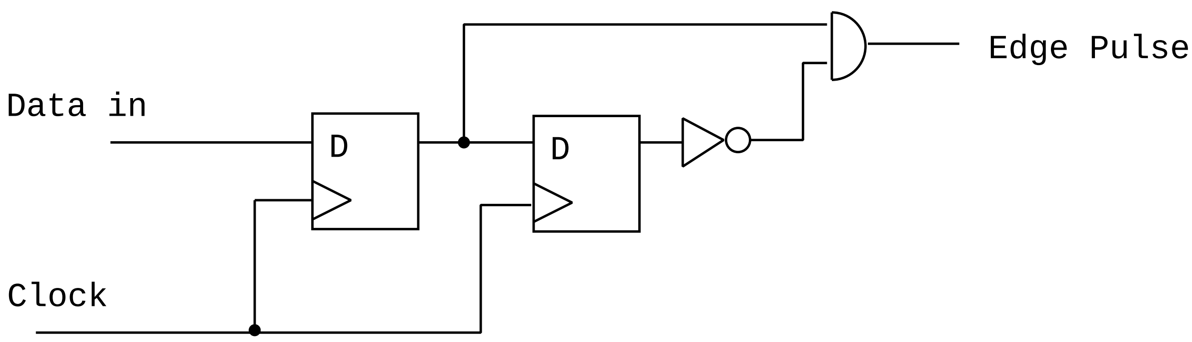
Design partitioning: The Cambridge Fast Ring



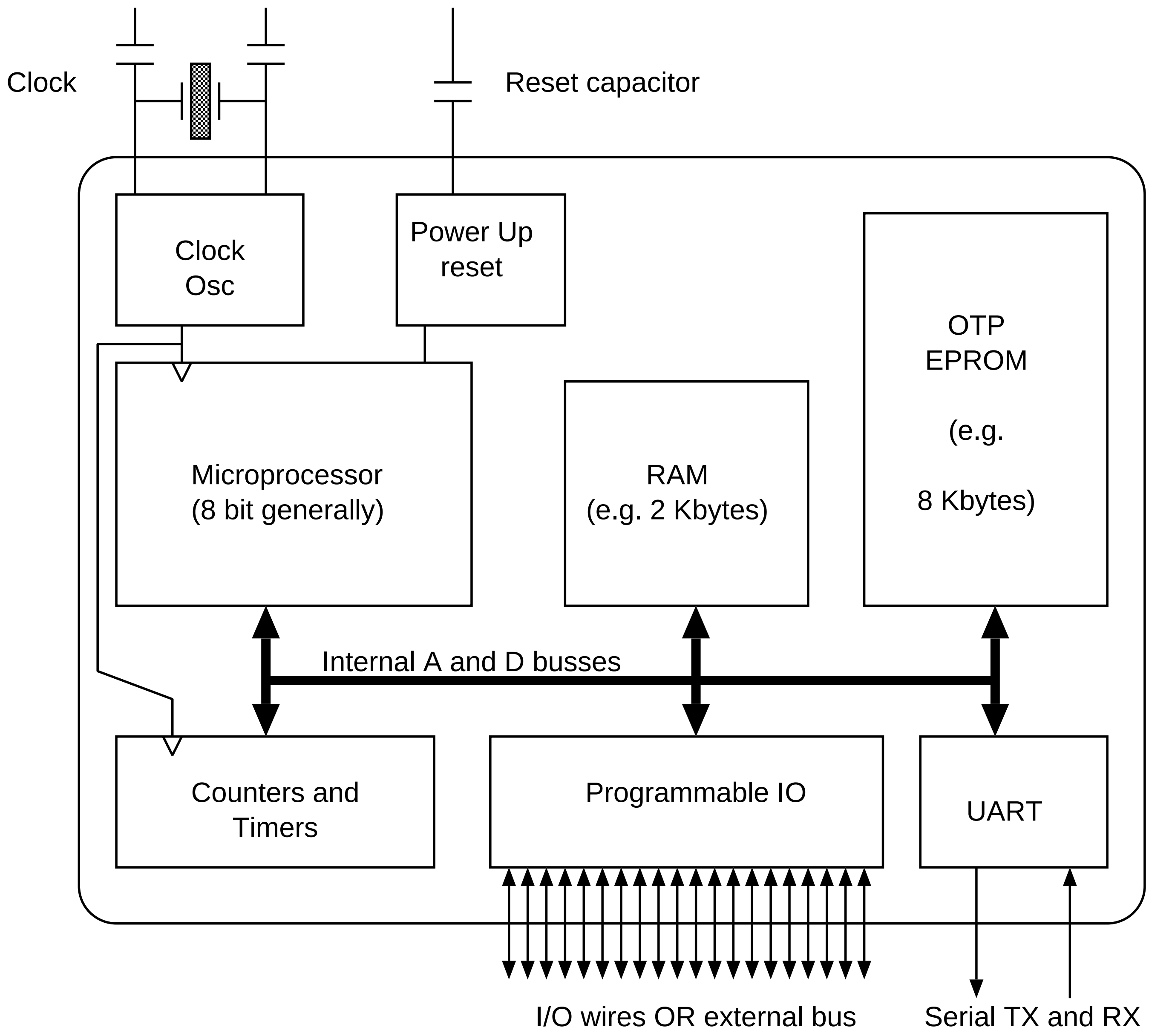
Design partitioning: An external modem



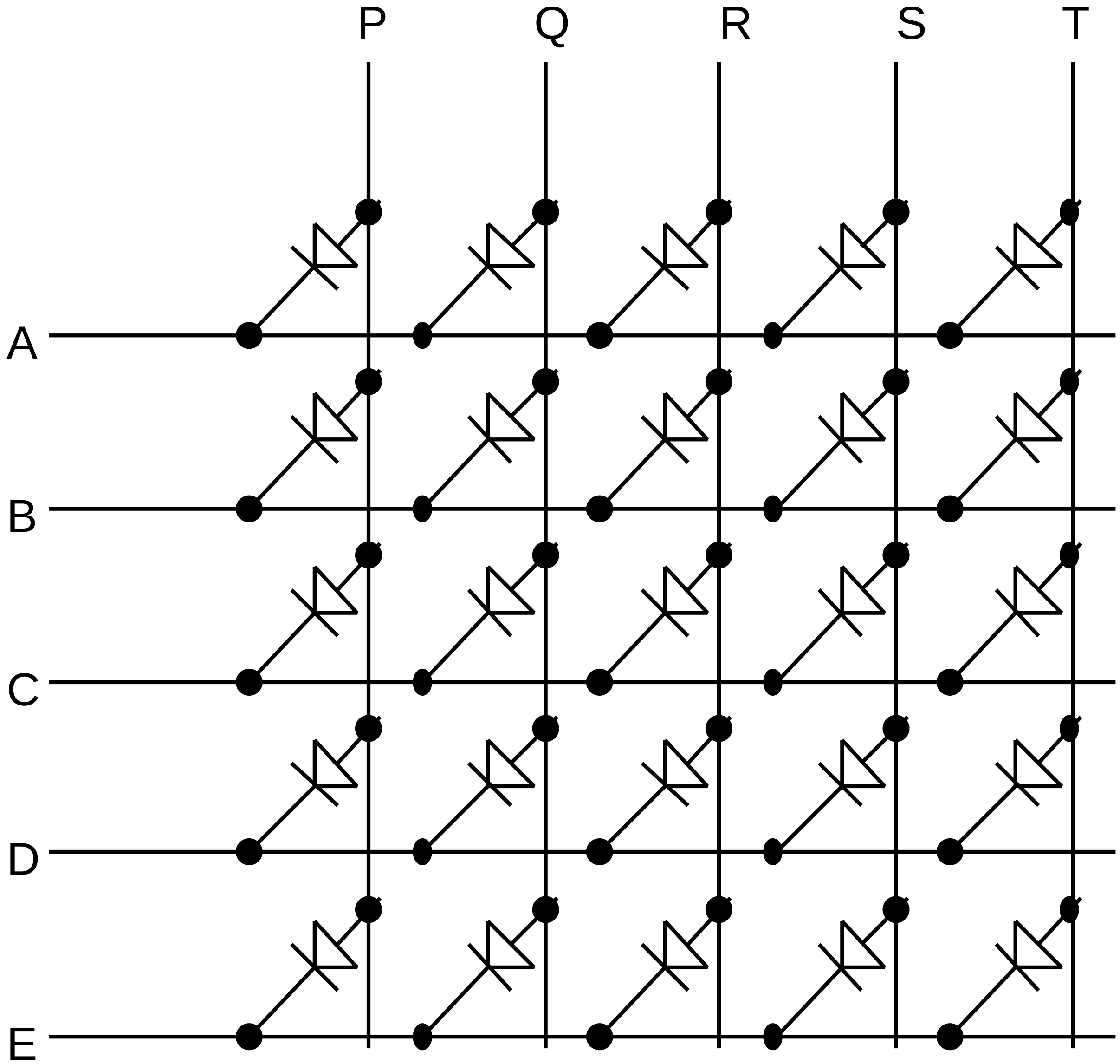
A simple edge detector



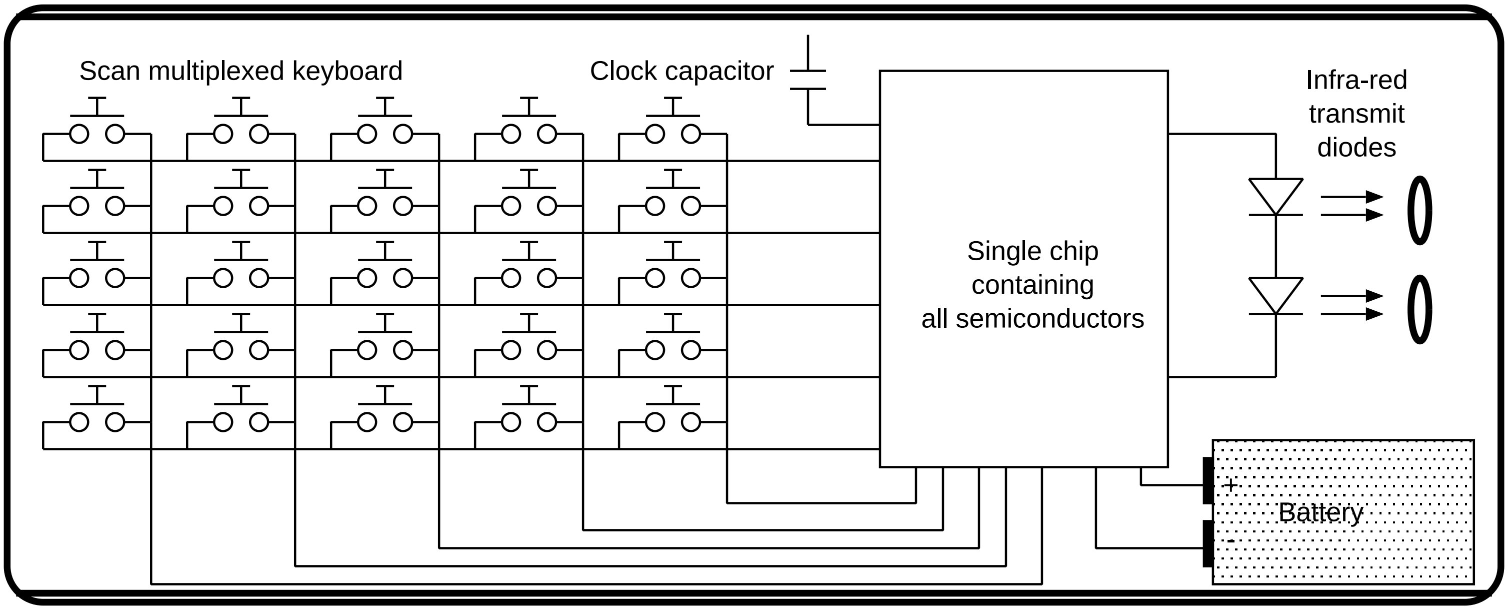
A Microcontroller



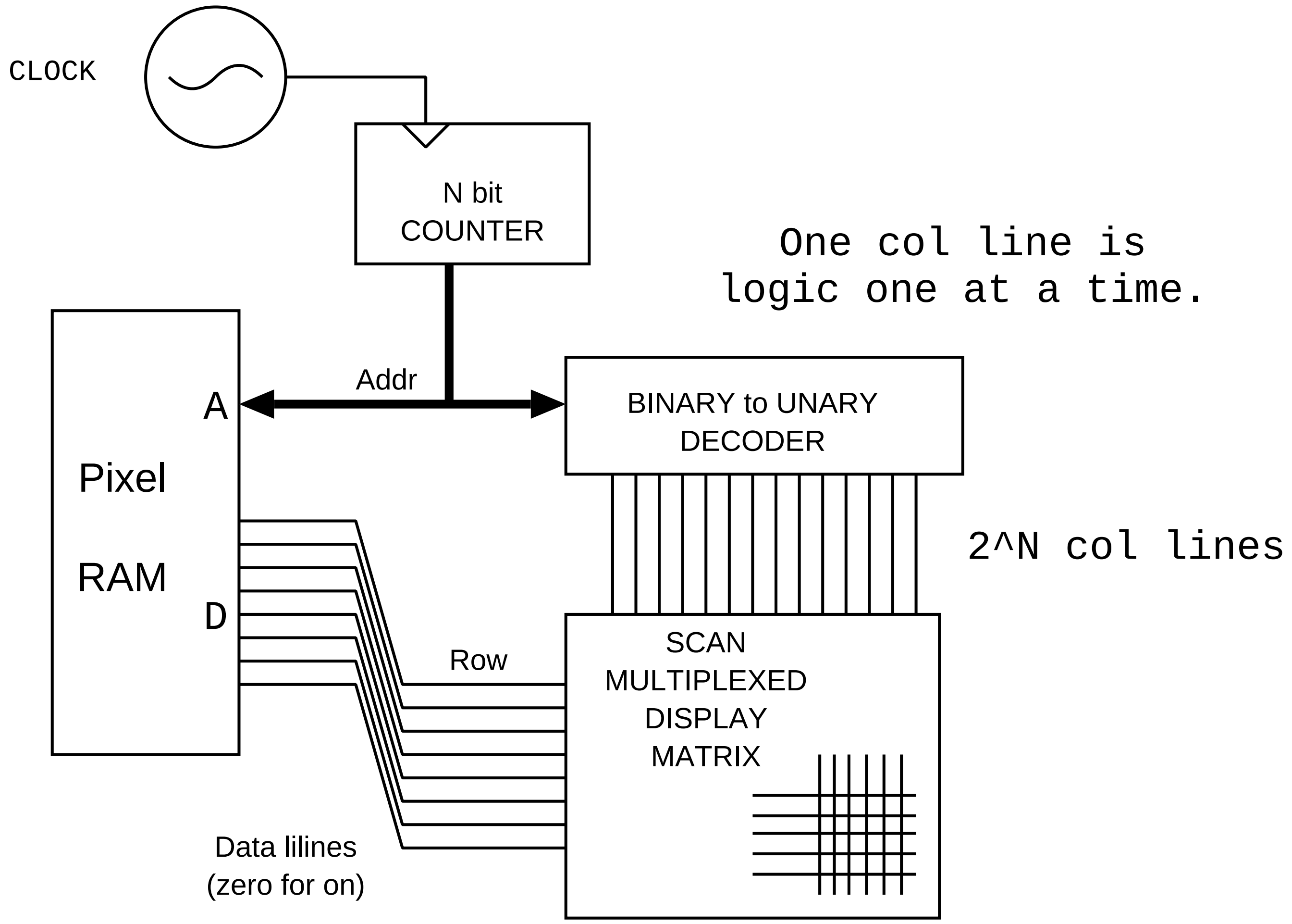
LEDs wired in a matrix to reduce external pin count



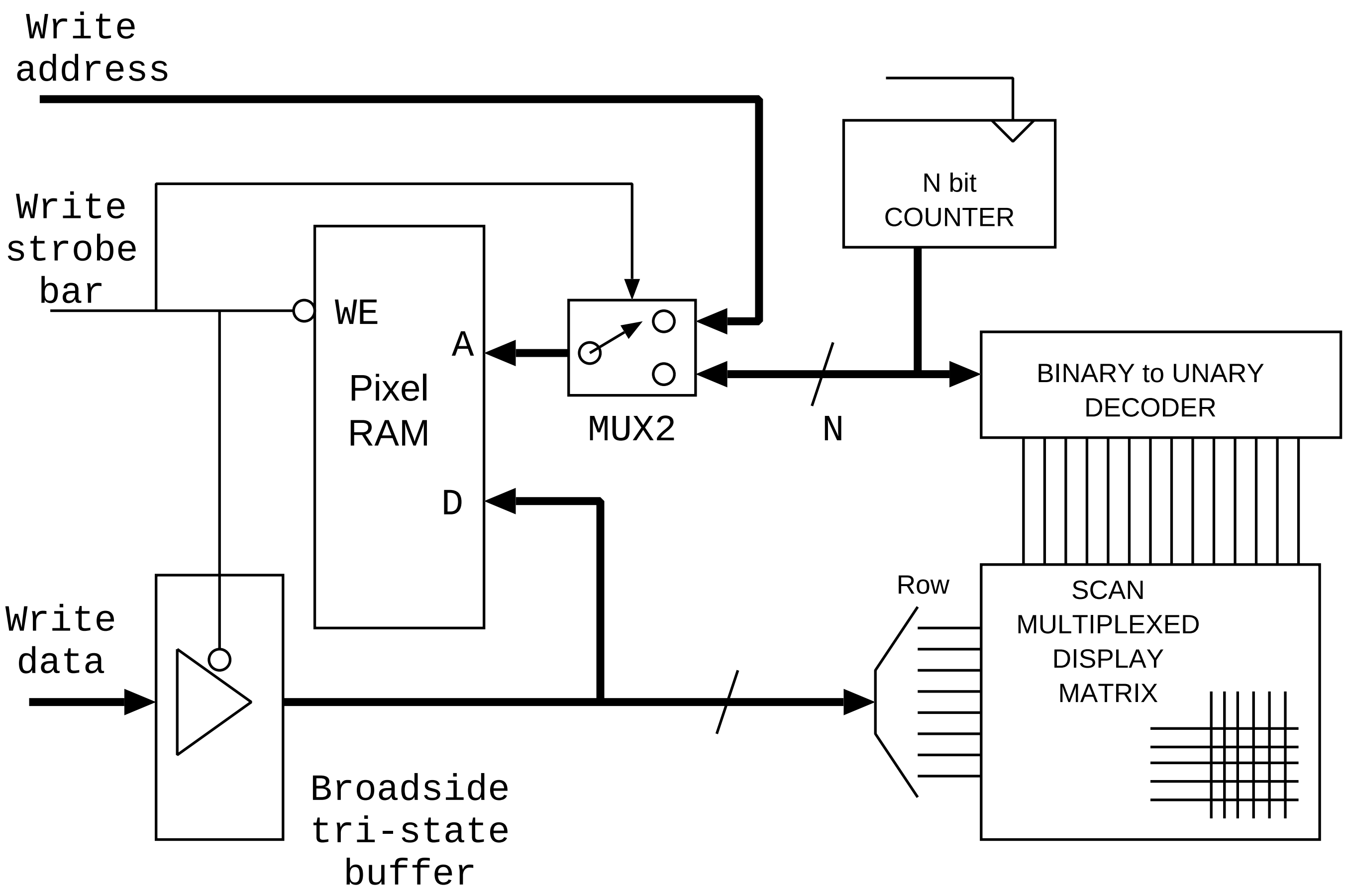
IR Handset Internal Circuit



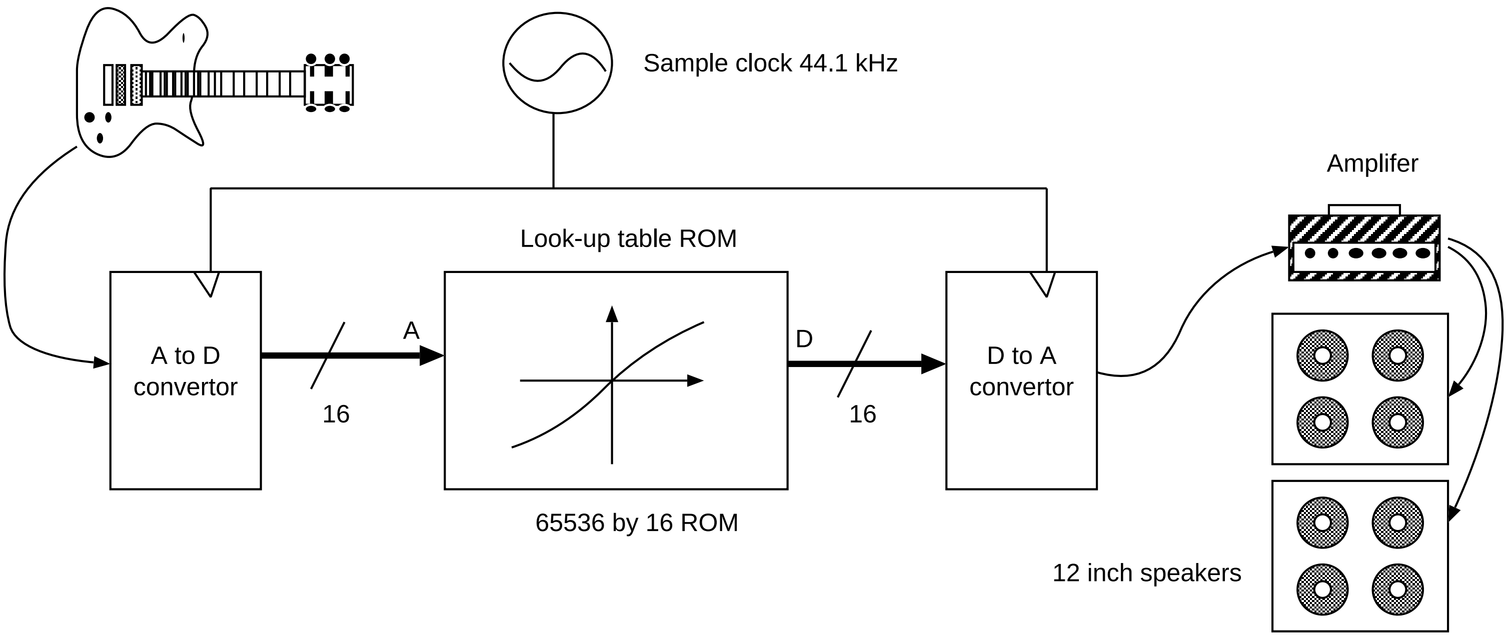
Scan multiplex logic for an LED pixel-mapped display



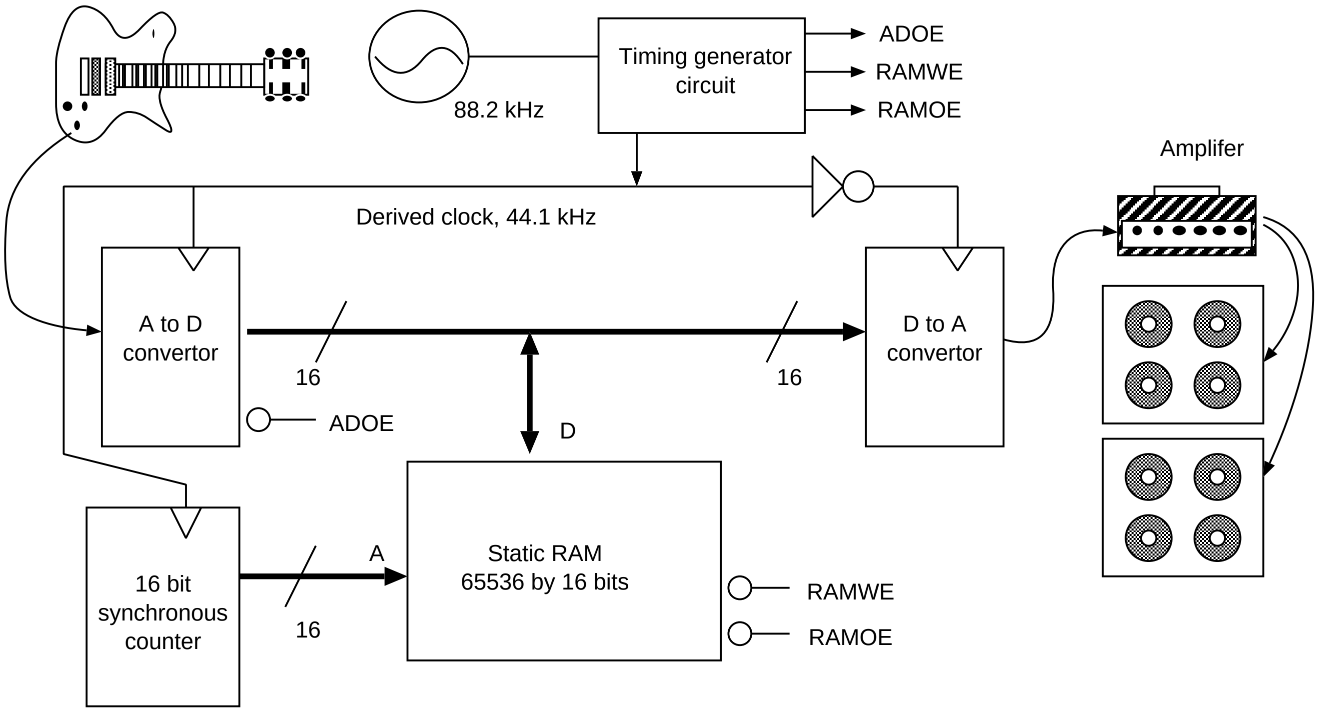
Addition of psudo dual-porting logic

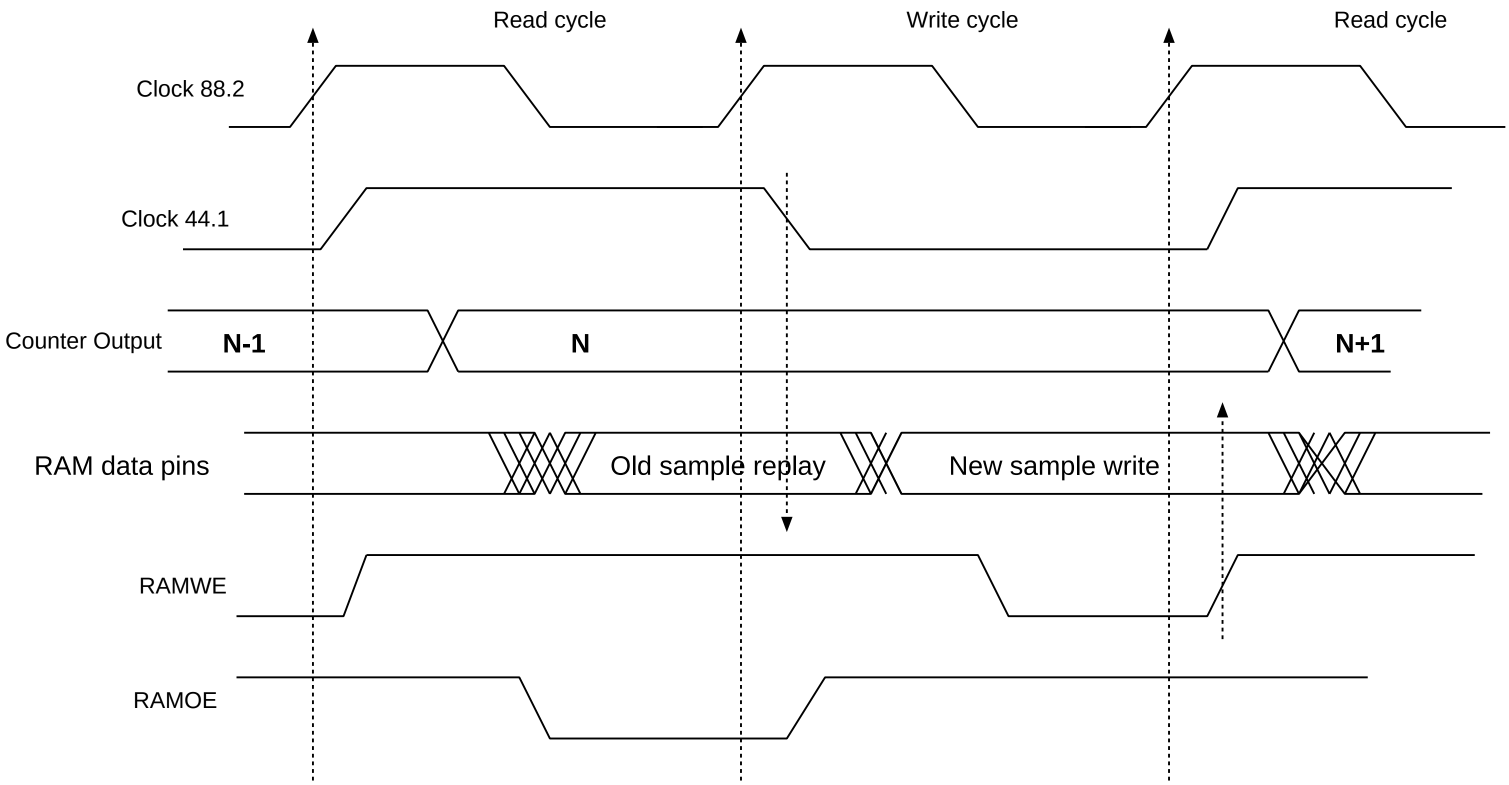


Use of a ROM as a function look-up table

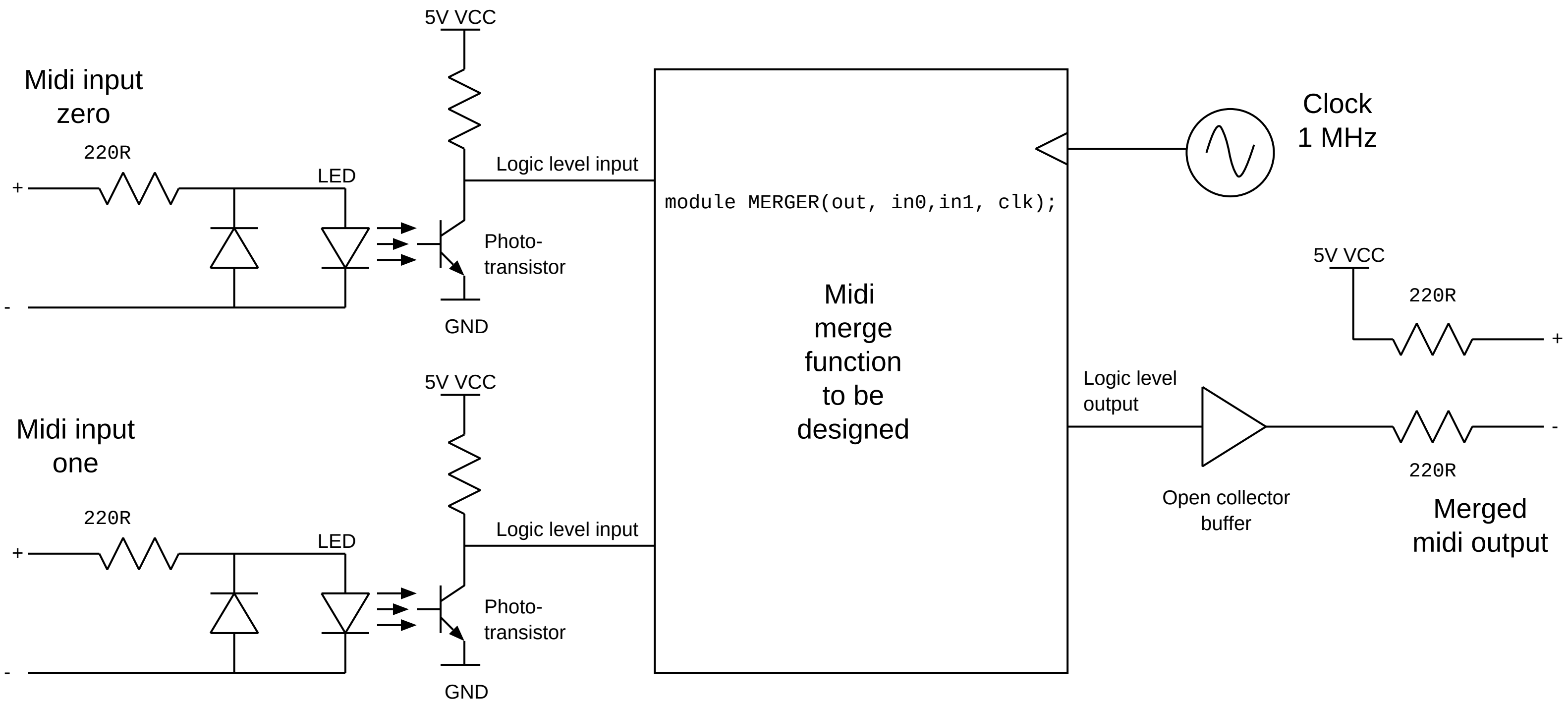


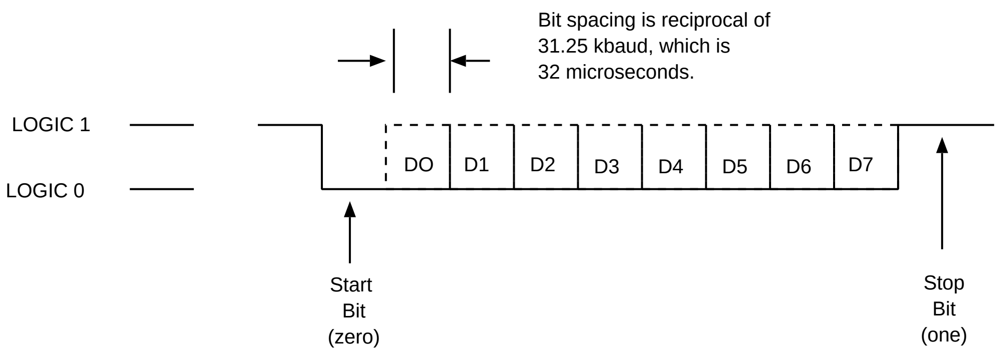
Use of an SRAM to make the delay required for an echo unit





Merge unit block diagram

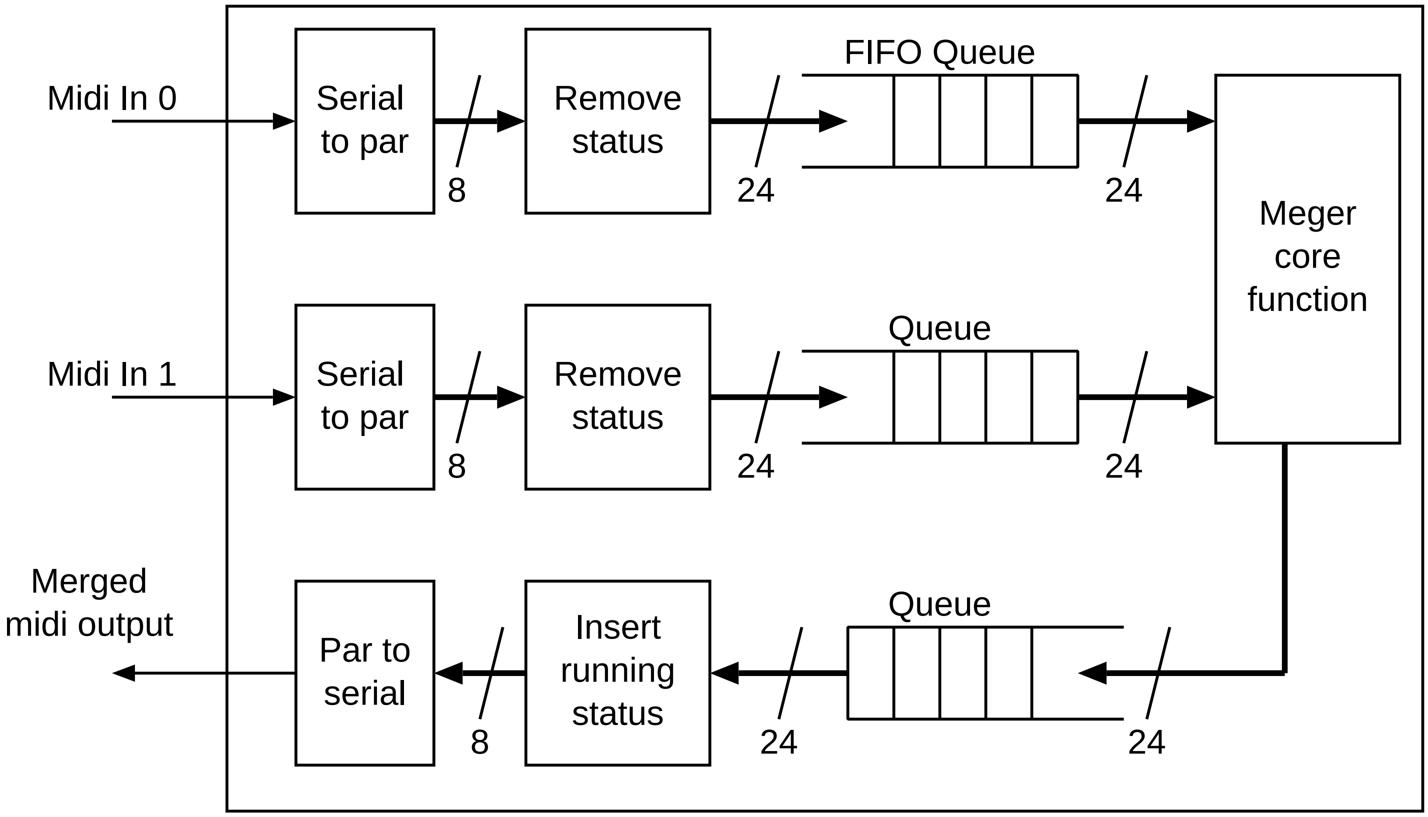




MIDI serial data format

9n kk vv	(note on)
8n kk vv	(note off)
9n kk 00	(note off with zero velocity)

MIDI merge unit internal functional units



The serial to parallel converter:

```
input clk;
output [7:0] pardata;   output guard;
```

The running status remover:

```
input clk;
input guard_in;   input [7:0] pardata_in;
output guard_out; output [23:0] pardata_out
```

For the FIFOs:

```
input clk;
input guard_in; input [7:0] pardata_in;
input read; output guard_out;  output [23:0] pardata_out;
input read; output guard_out;  output [23:0] pardata_out;
```

For the merge core unit:

```
input clk;
input guard_in0; input [23:0] pardata_in0; output read0;
input guard_in1; input [23:0] pardata_in1; output read1;
output guard_out; output [23:0] pardata_out;
input read; output guard_out;  output [23:0] pardata_out;
```

Status inserter / parallel to serial converter are reverse of reciprocal units

Network Camera Node