QBERT Chip Report

David Harris

December 8, 2001

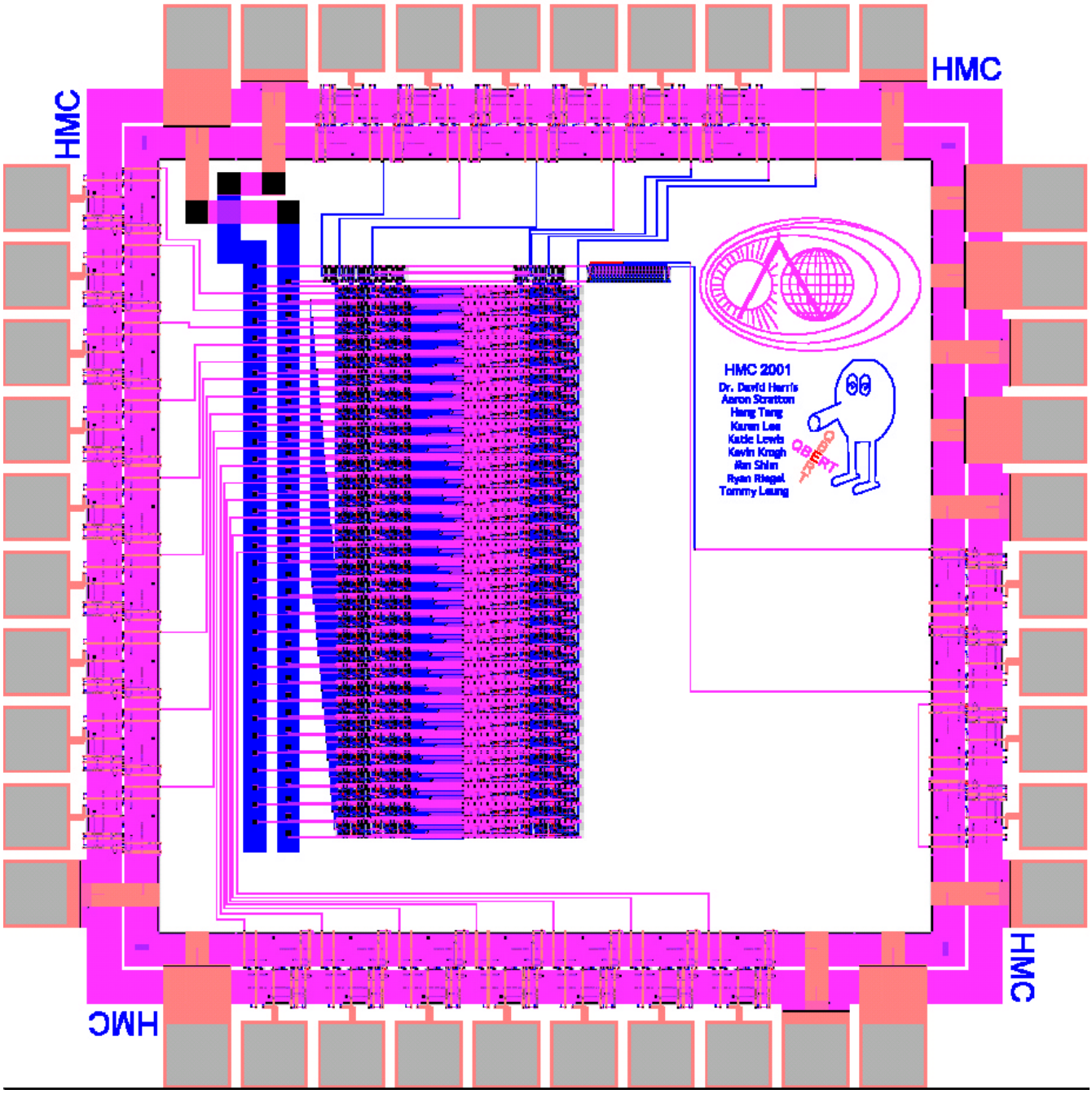


Table of Contents

Introduction

The Aerospace Corporation 2001-2002 Harvey Mudd College Clinic Team is building a 10 Gb/s Bit Error Rate Tester (BERT) composed of 16 interleaved 622 Mb/s links. The clinic team asked the Freshman Digital Electronics and Chip Design seminar to develop a proof of concept receiver for the BERT using a custom integrated circuit.

This report documents the custom chip, named QBERT. QBERT was developed by six freshmen in the seminar: Kevin Krogh, Karen Lee, Tommy Leung, Katie Lewis, Ryan Riegel, and Min Shim. They were assisted by trusty lab assistant Aaron Stratton and honorary freshman Hang Tang. Shamik Mitra and William Durley from the Aerospace clinic team provided specifications and technical assistance.

Beanstalk is implemented on a MOSIS 2.2x2.2 mm TinyChip in the AMI 0.5 micron process ($\lambda = 0.3$) in a 40-pin package. It was developed using the Electric CAD suite. Thanks to Steve Rubin for speedy bug fixes and tool enhancements.

Specifications

QBERT receives 16 bits of a pseudo-random bit sequence each cycle at a fast clock rate f_f. It collects the data into 32-bit chunks that are processed at a slower rate $f_s = ½ f_f$. The slower rate simplifies design and permits processing sequences with seeds up to 32 bits in length. For more information about pseudo-random bit sequences and the overall BERT architecture, see the Aerospace Clinic 2001-2002 Midyear Report.

The pseudo-random bit sequence has an *N*-bit seed and thus a length of 2^N-1. The QBERT chip receives sequences for N=31 or N=23, depending on input n_Sel being 1 or 0, respectively. When the Load_Pat input is asserted high, the chip latches the next 32 bits of input. When Load_Pat is released, the chip predicts the next 32 bits of input using the current 32 bits. If there is a discrepancy between the prediction and the actual incoming sequence, QBERT sinks current from the Eout line proportional to the number of mismatches.

Inputs	Outputs
D15 … D0	Eout
n_Sel	
Load_Pat	
F_PH1, F_PH2, S_PH1, S_PH2	

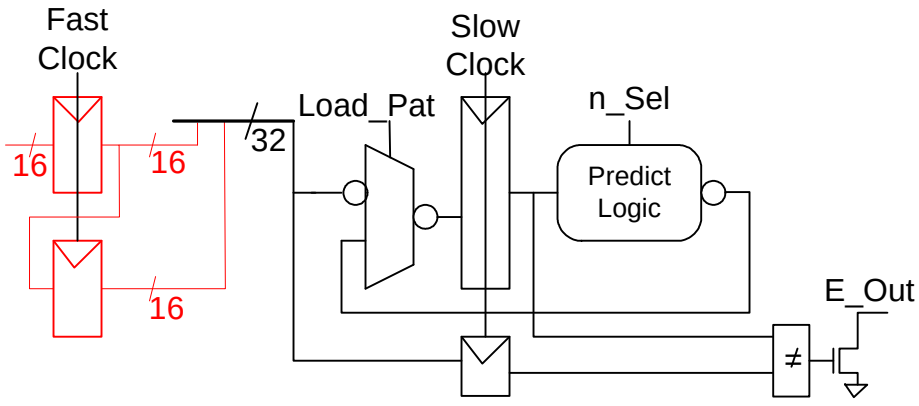
The clocks are provided as two-phase nonoverlapping pulses to avoid race errors. In the final system, the fast clock F will operate at 622 MHz and the slow clock S will operate at 311 MHz. In this prototype, the operating speed has not been determined but is expected to be lower, limited in particular by the frequency response of the 40-pin DIP package.

The QBERT chip also includes test structures to verify the pads and basic. The test structures include an enableable ring oscillator, and an input connected directly to an output pad. They have the following signals:

Inputs	Outputs
RingEN (1 = ring oscillator enabled, 0 = disabled)	RingOut (21 stage ring oscillator)
TestIn	TestOut (same as TestIn)

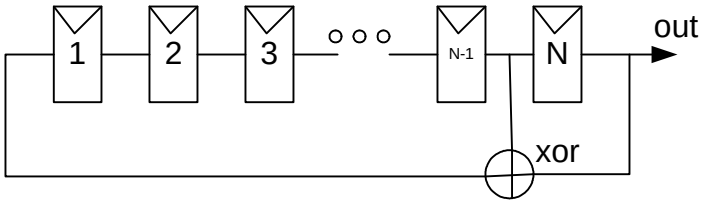
Theory of Operation

The QBERT datapath is shown below. On each fast clock cycle, 16 bits of data are received. Two chunks of data are combined to form a 32-bit word. When Load_Pat is asserted, the data is loaded into a prediction register. On subsequent slow cycles, the next sequence is predicted from the current one. The prediction algorithm depends on n_Sel. On each slow cycle, the actual input is compared against the predicted value. Mismatches cause open-drain transistors connected to E_Out to turn on. The number of mismatches is proportional to the current sunk at E_Out.



The datapath is constructed from a stack of identical horizontal bitslices. Sixteen vertical wires on the left side are used to route the data between the fast registers. Thirty two vertical wires distribute the current state to the predict logic to allow predicting the next state.

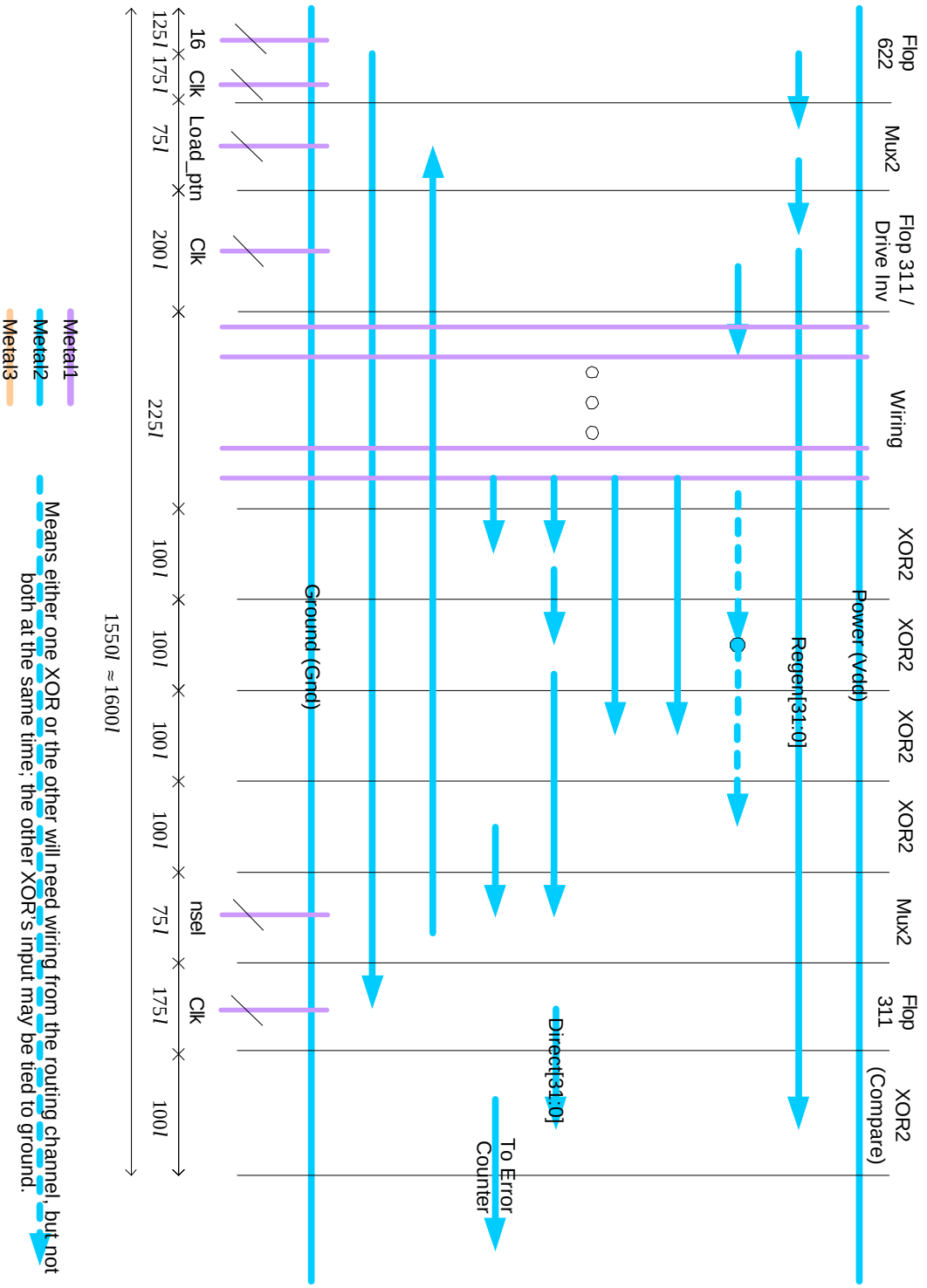
The PRBS is equal to that generated by a linear feedback shift register with the last two registers XORed and fed back to the input as shown in the figure below. Based on this, one can determine given the last 32 bits in the PRBS what the next 32 bits should be as the XOR of particular bits. The bits to combine are shown in the table below.



Predicted Bit	*N*=23	*N*=31
1	10, 11	2, 3
2	11, 12	3, 4
3	12, 13	4, 5
4	13, 14	5, 6
5	14, 15	6, 7
6	15, 16	7, 8
7	16, 17	8, 9
8	17, 18	9, 10
9	18, 19	10, 11
10	19, 20	11, 12
11	20, 21	12, 13
12	21, 22	13, 14
13	22, 23	14, 15
14	23, 24	15, 16
15	24, 25	16, 17
16	25, 26	17, 18
17	26, 27	18, 19
18	27, 28	19, 20
19	28, 29	20, 21
20	29, 30	21, 22
21	30, 31	22, 23
22	31, 32	23, 24
23	32, 10, 11	24, 25
24	10, 12	25, 26
25	11, 13	26, 27
26	12, 14	27, 28
27	13, 15	28, 29
28	14, 16	29, 30
29	15, 17	30, 31
30	16, 18	31, 32
31	17, 19	32, 2, 3
32	18, 20	2, 4

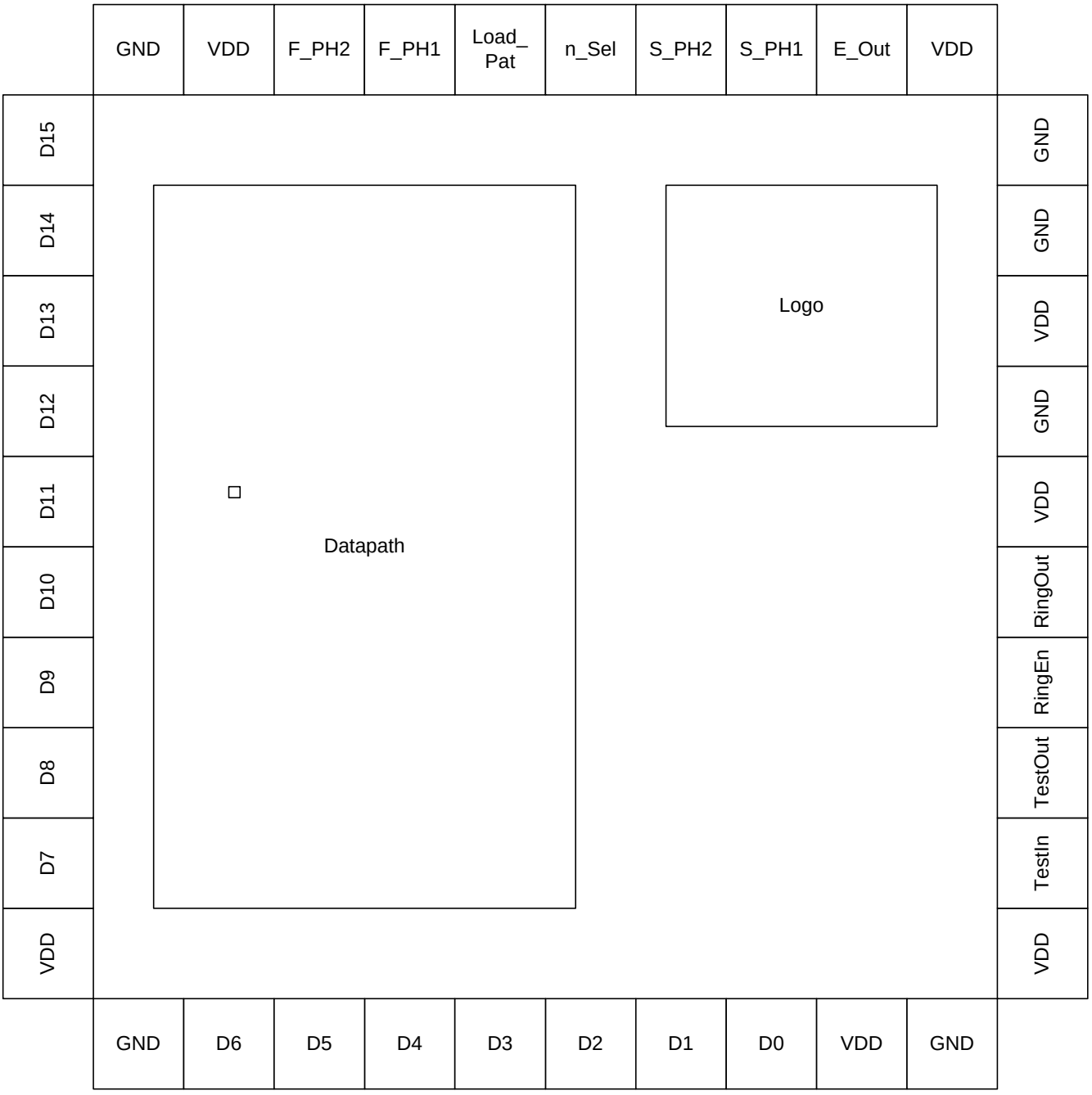
Bit Slice Floorplan

The following floorplan was prepared by the clinic team. It is slightly out of date and the inverter is in the wrong location. Refer to the schematics and layout for more information.



Pinout Diagram

The chip pinout is shown below.



Verification

The chip was verified in a number of ways.

A layout and schematic of the entire chip were created with the Pad Frame Generator. The generator placed pads off grid, so the pads had to be manually aligned to the grid.

They both were simulated with IRSIM. A weak PMOS transistor (width 1) with a grounded gate was tied to Eout in the layout and schematic for simulation purposes by adding the following line to the IRSIM deck:

```
p gnd E_Out vdd 2 1 33 26 g=S_vdd s=A_6,P_13 d=A_6,P_13
```

The simulation was invoked with the command:

```
irsim scmos100.prm qbertsch.sim  (or qbertlay.sim)
IRSIM> @ qbert.cmd
```

The simulation loads a bit sequence starting with 31 1's for *N*=31 and checks that the proper bit sequence is regenerated during the next 4 slow cycles. Then a new pattern with 23 1's for *N*=23 is loaded and the sequence is again regenerated for the next 4 cycles.

Network Consistency Check (NCC) was run with Ignore Power & Ground and Check Export Names. It passed both recursively and by expanding the hierarchy except that it flags the logo as not matching schematics.

Electrical Rule Check was run on the layout. It found the farthest distance from a contact to be about 184 for P-well and 238 for N-well. These long distances are on vdd/gnd pads with no transistors. The longest distance in the core is 90, and transistors are much closer. It found 2697 PMOS transistors and 2573 NMOS transistors.

Design Rule Check was run on the entire layout with the logo deleted. It passed with three levels of metal in the moscmos process with stacked vias and alternate contact rules.

A cif file was exported with the default settings. There were some resolution errors reported. This is not fatal, but impacts checking with other programs like Magic.

At the time of this report, the CIF file has not been checked in Magic.

Testing Results

The chip successfully passed all the tests in qbert.cmd. When errors were introduced, the error output current measured across a resistor grew linearly for 0, 1, and 2 errors. The ring oscillator was too fast to directly measure on a 100 MHz scope.

Summary of Files

The master files needed to recreate the project are on Prof. Harris' NT machine at:

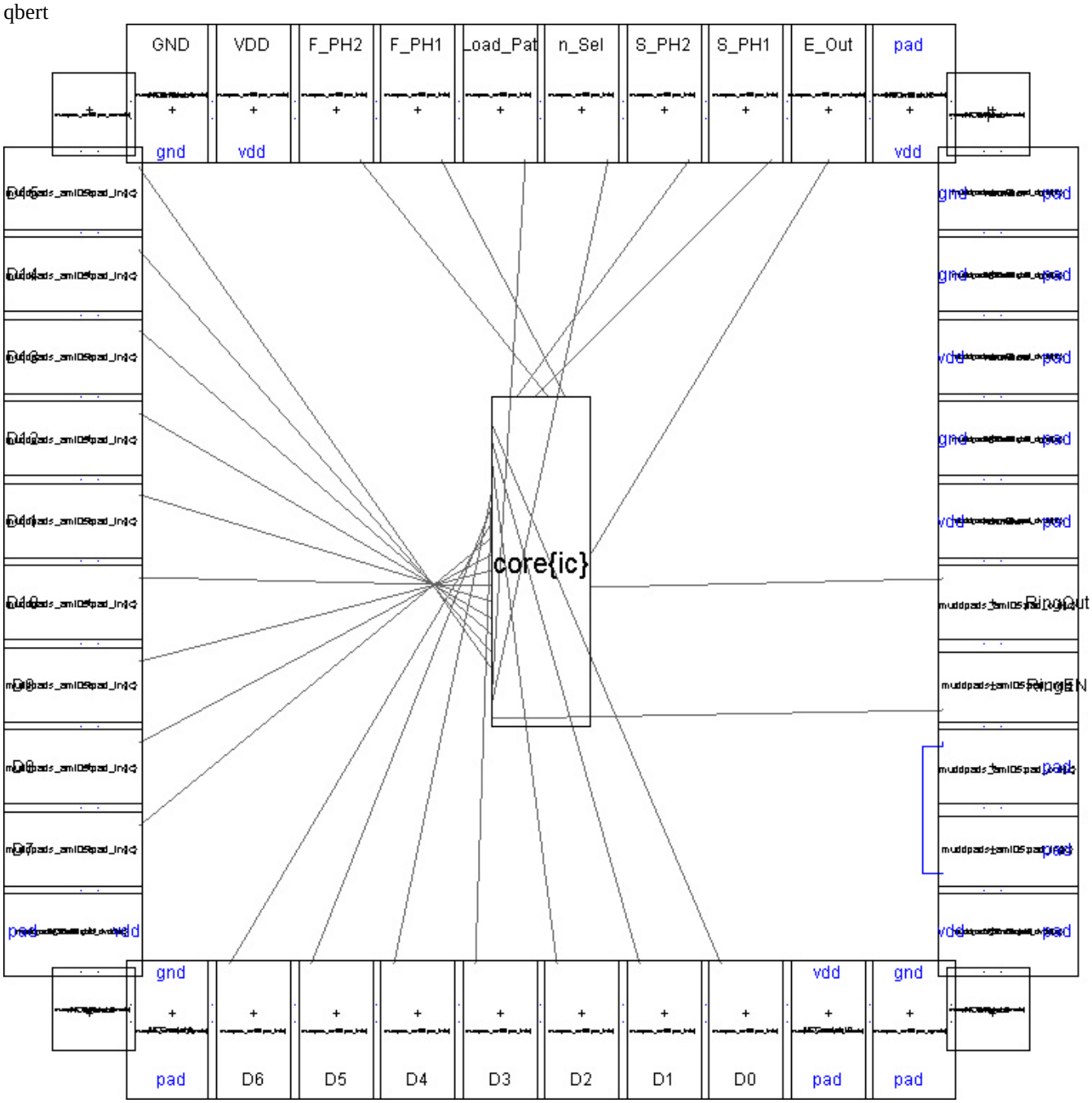
C:\Classes\chipseminar\Fall 2001\QBERT

They include:

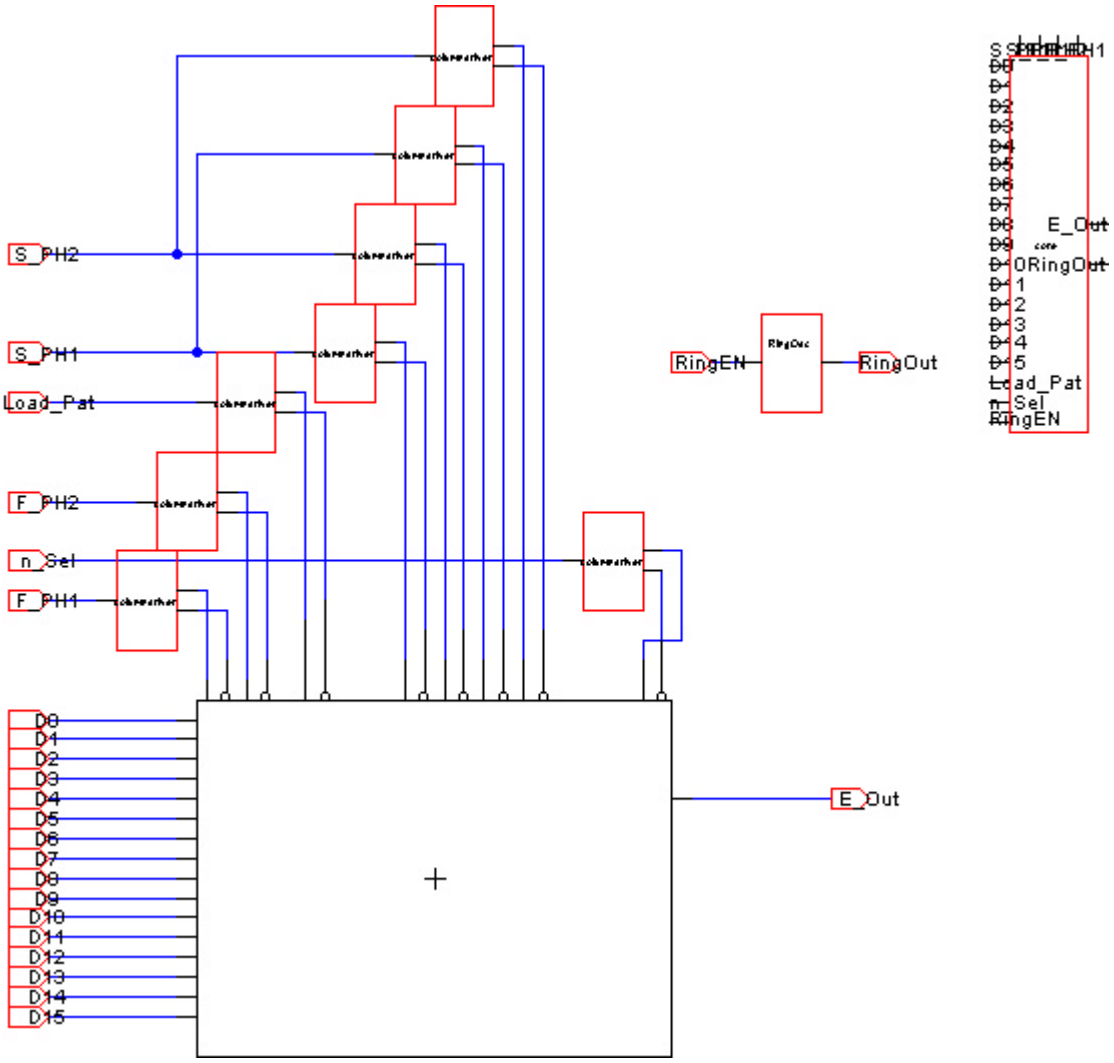
qbert.elib:	the top-level design
muddpads_ami05.elib:	pad frame with HMC logo
qbert.cif:	master CIF file
qbert.cmd:	IRSIM command file to test qbert
qbertlay.arr:	pad arrangement file for layout
qbertsch.arr	pad arrangement file for schematic

Schematics

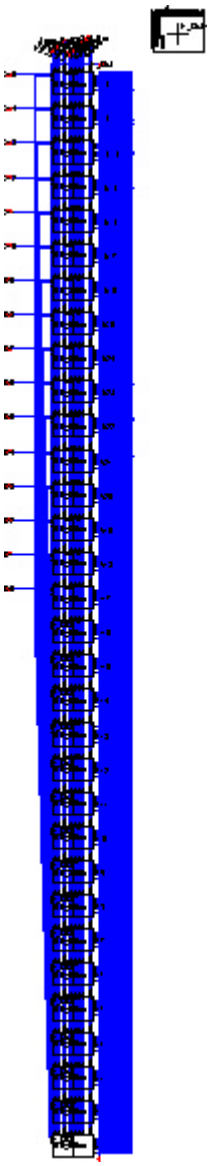
qbert



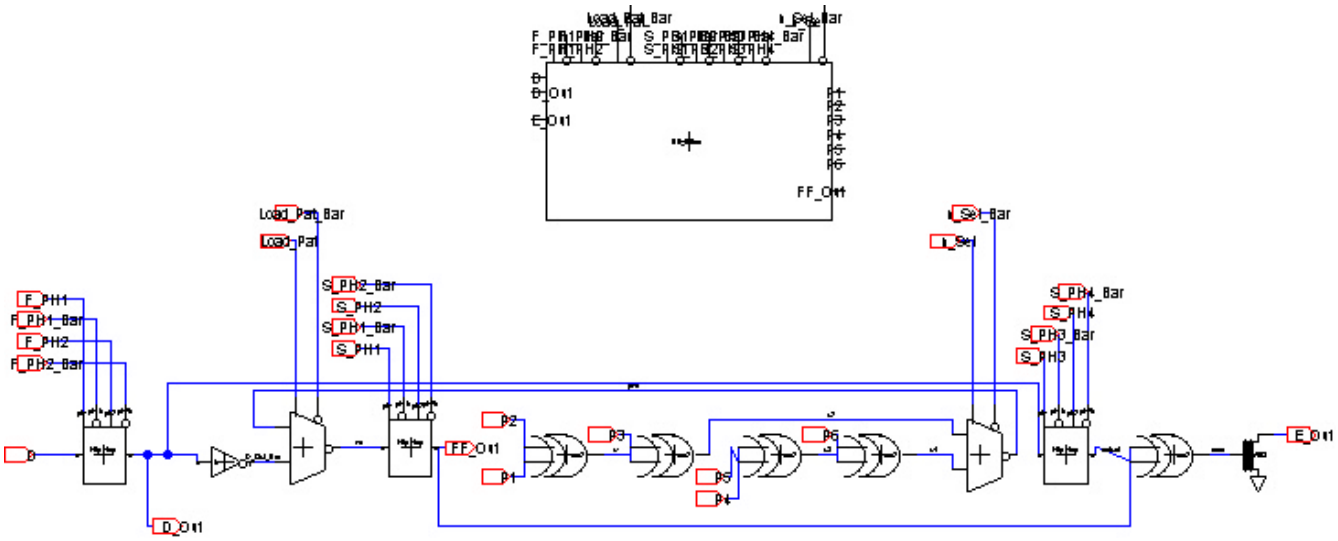
core


S_PH2
S_PH1
Load_Pat
F_PH2
n_Sel
F_PH1
D0
D1
D2
D3
D4
D5
D6
D7
D8
D9
D10
D11
D12
D13
D14
D15
E_Out
RingEN
RingOut
E_Out
RingOut
Load_Pat
n_Sel
RingEN

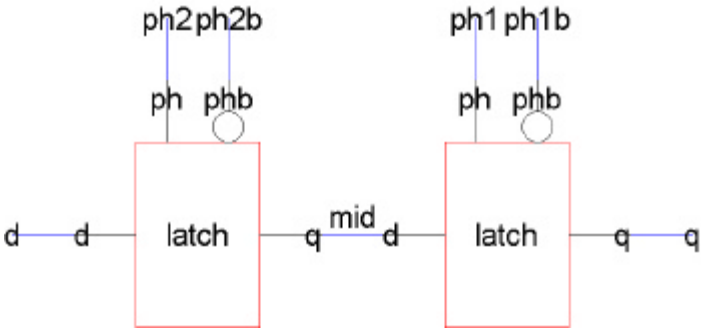
datapath



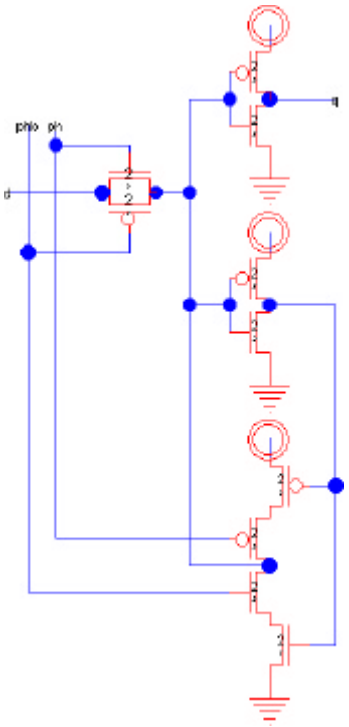
bitslice



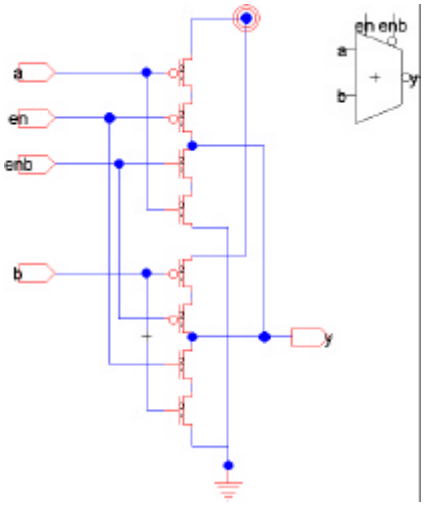
flop



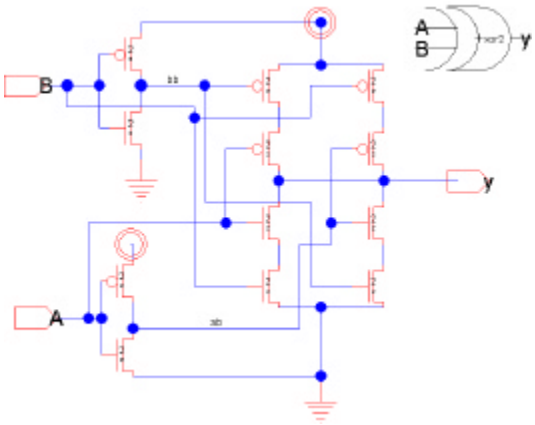
latch



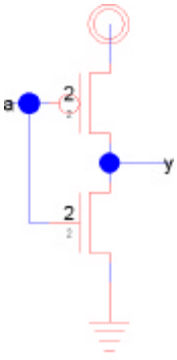
mux



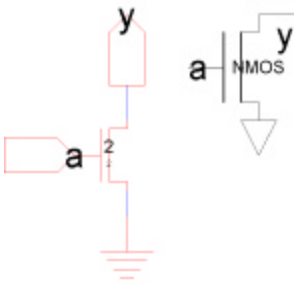
xor



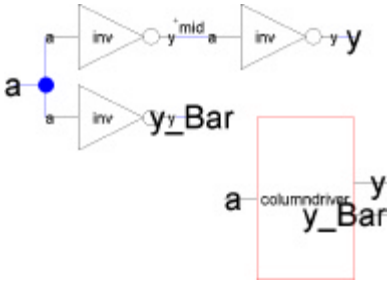
inv



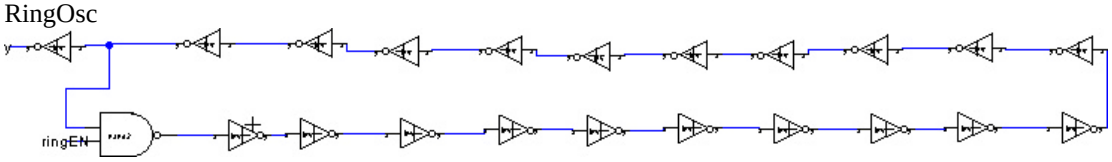
NMOS



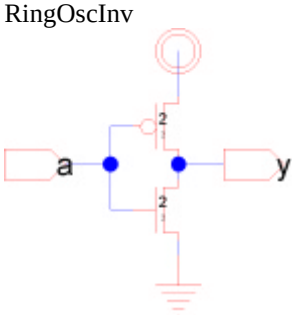
columndriver



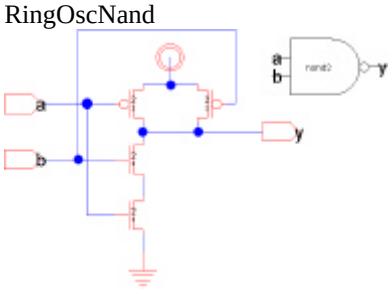
RingOsc



RingOscInv

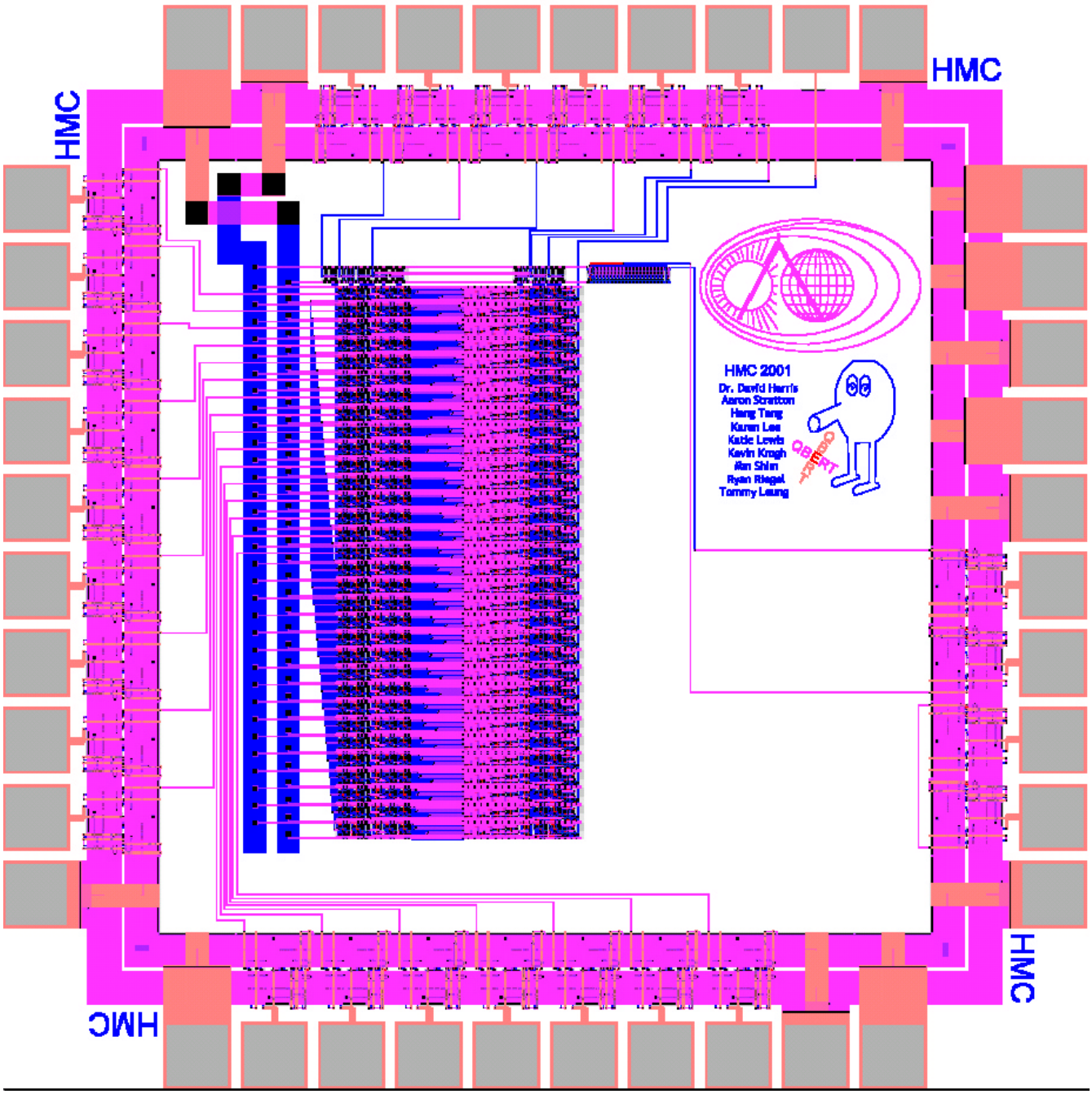


RingOscNand

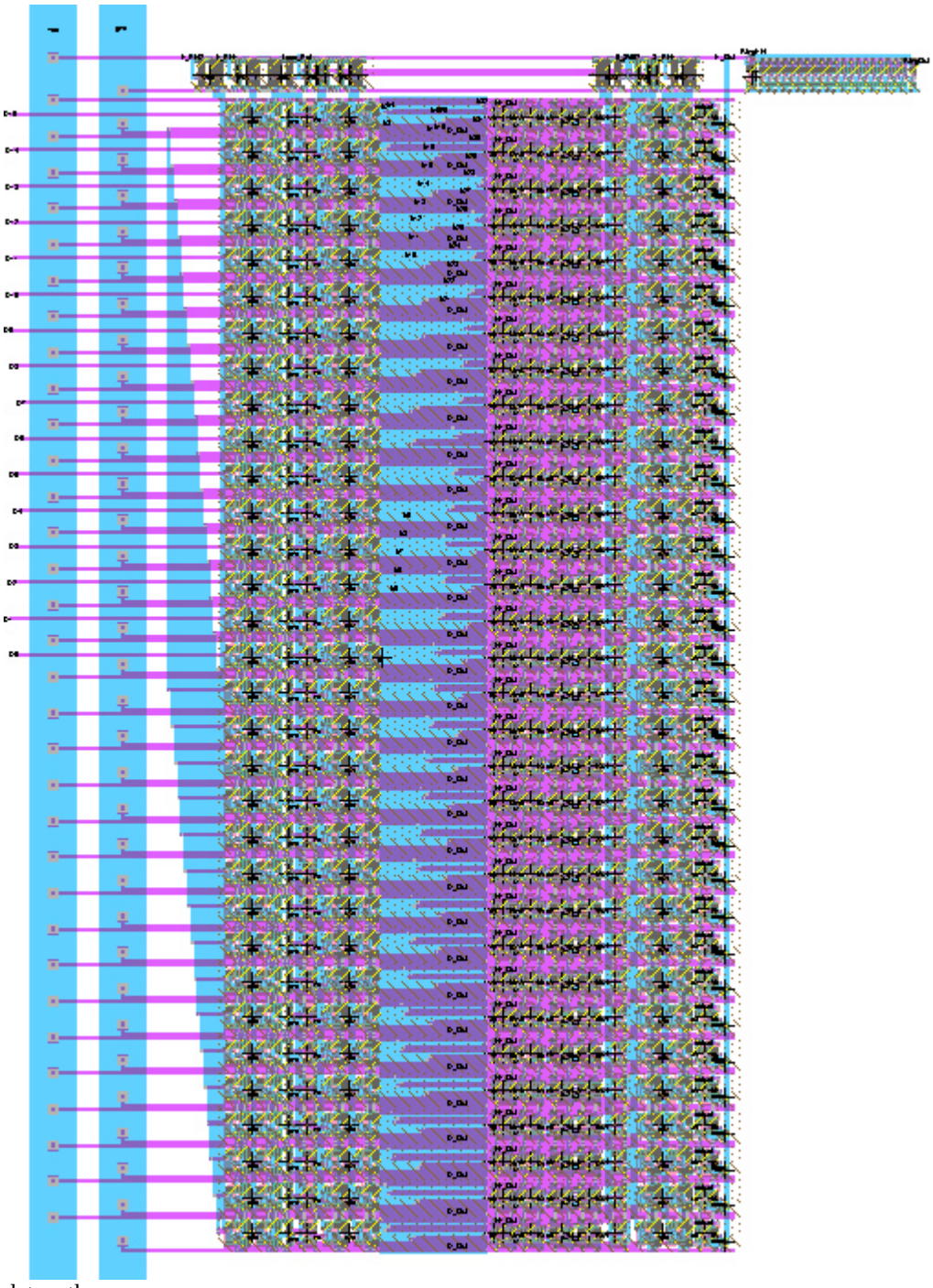


Layout

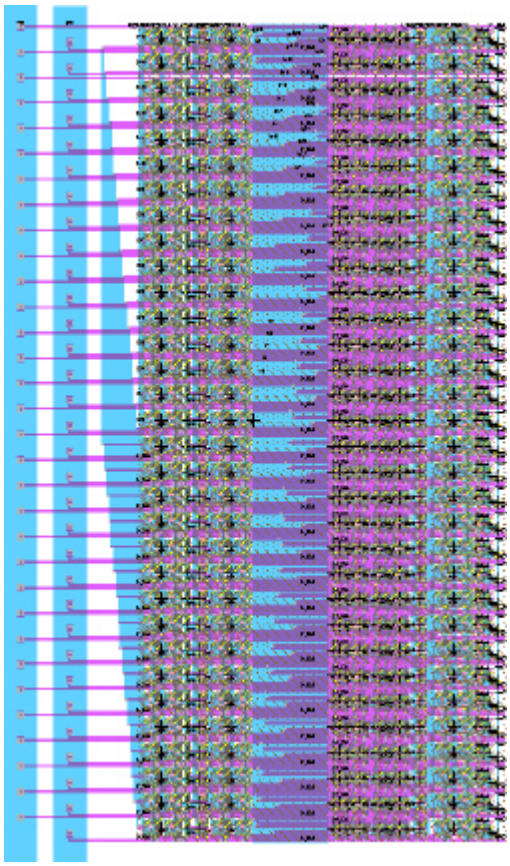
qbert



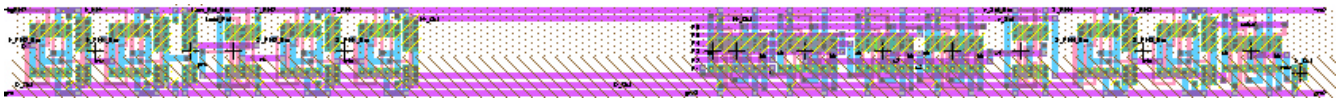
core



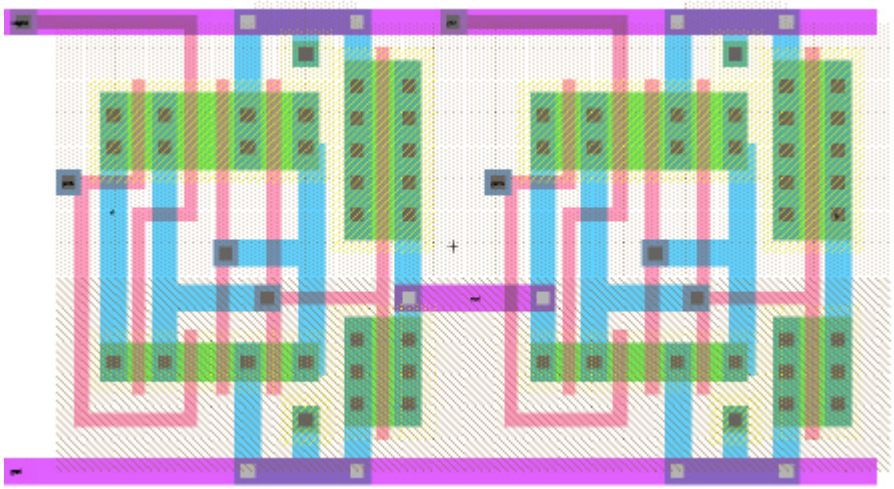
datapath



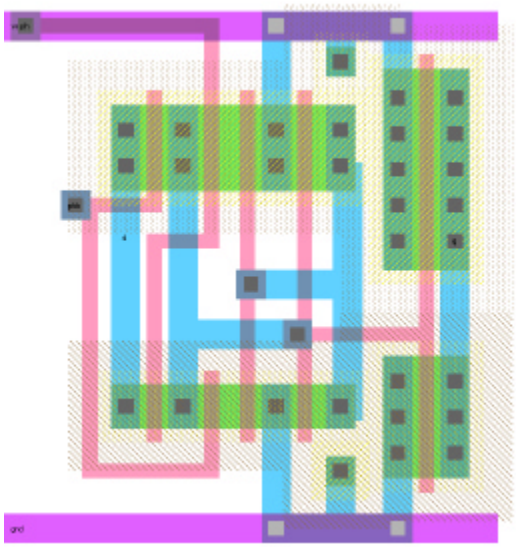
bitslice



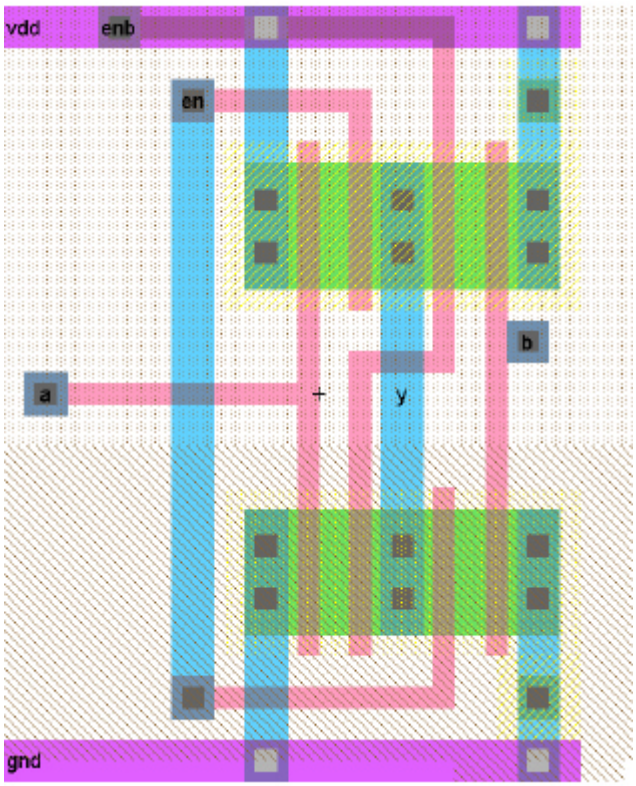
flop



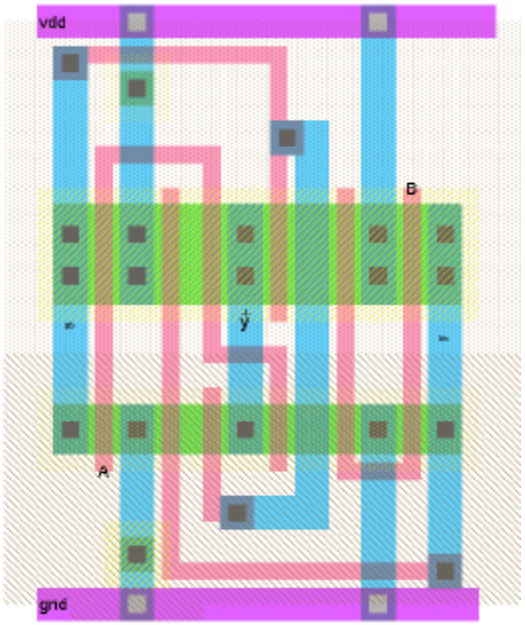
latch



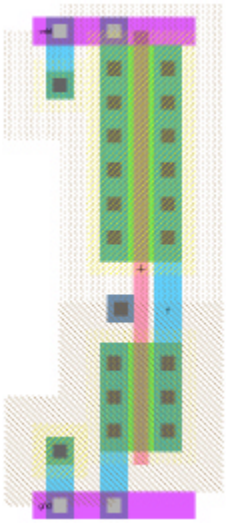
mux



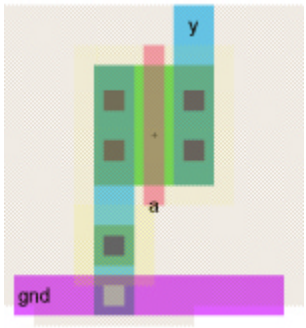
xor



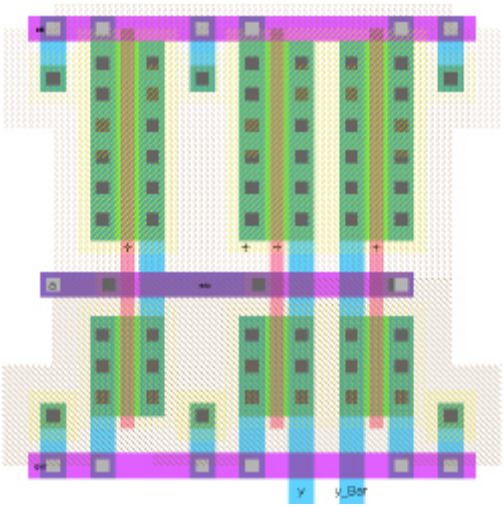
inv



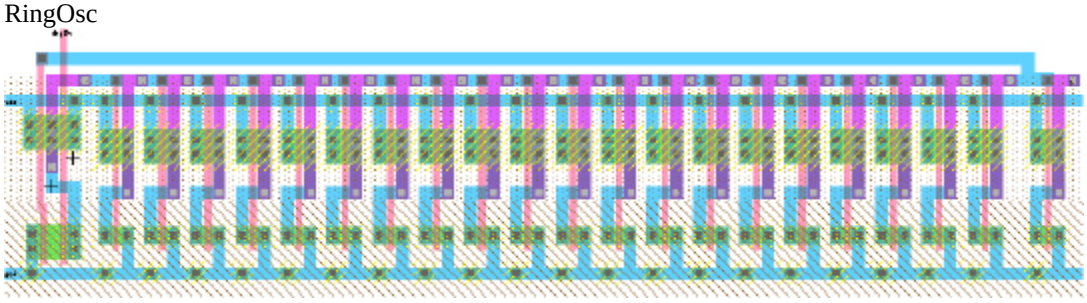
NMOS



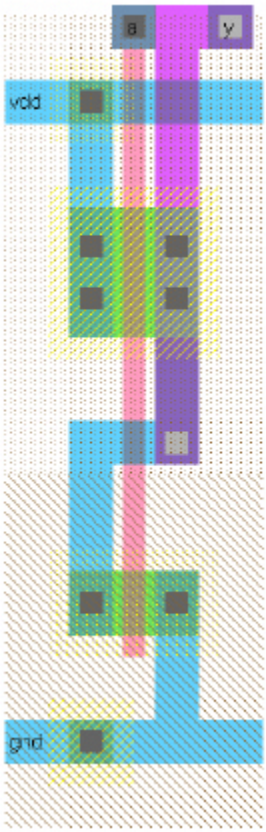
columndriver



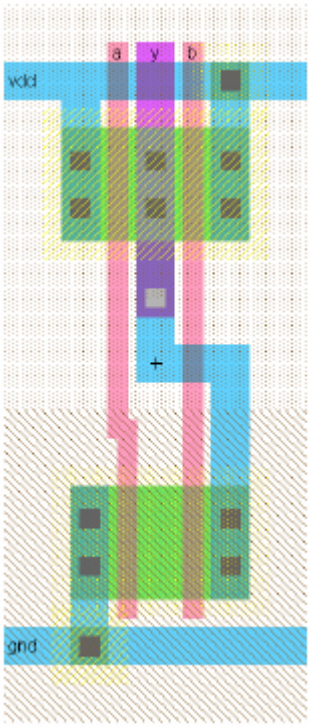
RingOsc



RingOscInv



RingOscNand



logo