A Broadside Register
A broadside two-to-one multiplexor
A dual port register file
The ROM takes $A$ address bits named $A_0$ 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 $A_0, A_1, A_2, A_3, A_4$ and the data lines $D_0, D_1, ..., D_7$.

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.

Binary to unary decoder

Address Input
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 $4^A \times D$. 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 to be given and multiple column 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 extended 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

Outside the chip

Inside the chip

PLL Circuit

VCO

264 MHz

Divide 8

Clock distribution tree

External clock input

33 MHz

Power-on reset

Supply

Active low
Reset output

Vo

Vi

C

Ground
Driving a heavy current or high-voltage load
Debouncer circuit for a two-pole switch

[Diagram showing a debouncer circuit with a switch, pullup resistors, and logic gates to debounce the switch's bounces.]

Switch

Gnd

+5Volt supply rail

Pullup Resistors

Output

Bounces

A

B

Output

(Chart showing waveforms for A, B, and Output to illustrate debounce process.)
ALU and flags register

Function Code

4

N

A-input

N

B-input

N

ALU

C

N

Z

V

Flags Clock

Carry In

Output

Flags register

Flags Clock
ALU and register file

Clock source

A-input

B-input

Register file
16 registers
of 8 bits

4 bit counter

FUNCTION GEN
for F code

FUNCTION GEN
for A input

4 bit

Zero defect

4

8

FUNCTION GEN
for A input

D

8

4

A-input

8

A

8 bit ALU

Function Code

B

Output

Din

D

8 bit

Output

Carry In

Carry Out

8
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

(Micro-)Processor

Register File (including PC)

Control Unit

Execution Unit + ALU

Memory

Static RAM

16 kByte

Address bus (16 bits)

Data bus (8 bits)

Clock

Reset

R/Wb

A15

A14

A13

Memory Map decoder circuit

Often a ‘PAL’ single chip device.

1 K Byte ROM

Read Only Memory

A0-9

Enb

RAM_ENABLE_BAR

A0-13

R/Wb

R/Wb

ROM_ENABLE_BAR

UART Serial Port

UART_ENABLE_BAR

Rs232 Serial Connection

16
PC Motherboard, 1997 vintage

- SIMM 4
- SIMM 3
- SIMM 2
- SIMM 1
- COM1
- COM2
- USB
- IDE-1
- IDE-2
- Floppy
- BIOS ROM
- CACHE RAM
- PCI1
- PCI2
- PCI3
- ISA 16 BIT SLOTS
- Main memory DRAM
- PSU
- KY BD
- PRINTER
- Pentium CPU
- CACHE Control
- General glue
- BATTERY
- Clock
- Regulator
Parallel Port Interface Logic

Address Data

Strobe Acknowledge

Parallel Data Busy

Valid Data For Transfer To Peripheral Device

Parallel Data

Strobe_bar

Acknowledge

Busy

Ready for next data

Flow control: New data is not sent while the busy wire is high.
Serial Port (UART)

Flow control: New data can be sent at any time unless either:
- additional signals are used to indicate clear to send
- a software protocol is defined to run on top (Xon/Xoff) by reserving certain of the bytes.
Keyboard and/or PS/2 port

PS/2 Keyboard/Mouse Cable
1. Clock
2. Ground
3. Data
4. Spare
5. Power +5Volts
6. Spare
Canonical synchronous FSM

[Diagram of a canonical synchronous finite state machine (FSM)]

- Inputs: $I_0, I_1, I_2, \ldots, I_{M-1}$
- Clock
- Moore Outputs
- Mealy Outputs
- Current State Feedback
- Loop-Free Combinatorial Logic Blocks
- State Flops: $Q_0, Q_1, Q_2$
Timing Specifications

Data in

Clock

Q output

Clock

Data in

Hold time

Setup time

Q output

Propagation delay
Typical nature of a critical path

Clock

D Q

A

B

C

D

D Q

Clock

A

B

C

D

Margin

Setup

Period = 1/F
Johnson counters

Clock

Q1 Q2 Q3

D QA
Q1

D Q2
Q2

D Q3
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

Clock enable

Data in

Clock

Q Output

LOGIC SYMBOL

AN EQUIVALENT CIRCUIT

Clock enable

Data in

1

0

Q Output

Clock
A Gated Clock

Enable expression
Enablebar

Master Clock

Synchronous subsystem requiring gated clock

D
J
K

29
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

Transmit clock domain

Guard signal

Command or info bus

N

RX clock

Receiving clock domain

Optional second D-type

TX clock
Paths between FSMs w/ derived clocks

Moore feedback to parent clock domain

Inputs

Clock Input

Moore

Inputs

FSM

Feedforward of outputs to son FSM

Moore Outputs

Mealy Outputs

FSM

Moore Outputs

Mealy Outputs

FSM

Inputs
Dicing a wafer

(Chips are not always square)
A chip in its package, ready for bond wires

IO and power pads
## Die cost example

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

Integrated Circuits

Standard Parts

Masked ASICs

Field Programmable Parts

Commodity Parts

Full Custom

Semi Custom

FPGA

Array Logic (PALs)

Standard Cell

Gate Array

General Chip Products
Field Programmable Gate Arrays
A configurable logic block for a look-up-table based FPGA

General inputs

Clock input

Combinatorial function generator

Programmable multiplexers

First output

Second output
A simple IO block FPGA

Connections to central array.

Tristate control

Programmable multiplexor

Output enable

Output buffer

Input buffer

Output

Input

Bond PAD
Contents of the PAL macrocell

- Input buffer
- Clock Net
- I/O Pad

- Feedback to array
- Output enable term
- Main input S-of-P

- D-type flip-flop
- Programmable multiplexer
- Tristate output pad
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 = \neg 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
xxxx xxxx xxxx xxxx xxxx xxxx xxxx
xxxx xxxx xxxx xxxx xxxx xxxx xxxx
(xmacrocell fuse)
Delay-power style of technology comparison chart

<table>
<thead>
<tr>
<th>Technology</th>
<th>Device</th>
<th>Propagation Delay (ns)</th>
<th>Power (mW)</th>
<th>Product (pJ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMOS</td>
<td>74hc00</td>
<td>7 ns</td>
<td>1 mW</td>
<td>7</td>
</tr>
<tr>
<td>TTL</td>
<td>74f00</td>
<td>3.4 ns</td>
<td>5 mW</td>
<td>17</td>
</tr>
<tr>
<td>ECL</td>
<td>sp92701</td>
<td>0.8 ns</td>
<td>200 mW</td>
<td>160</td>
</tr>
</tbody>
</table>

Lines of constant delay-power product
Logic net with tracking and input load capacitances

- Driving Gate
- Track to substrate capacitance proportional to total track length (area)
- Driven gates
- Parasitic input capacitance
An example cell from a manufacturer’s cell library

NAND4 Standard Cell
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)

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Outputs</th>
<th>O/P Falling</th>
<th>O/P Rising</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>(ps)</td>
<td>ps/LU</td>
</tr>
<tr>
<td>A</td>
<td>F</td>
<td>142</td>
<td>37</td>
</tr>
<tr>
<td>B</td>
<td>F</td>
<td>161</td>
<td>37</td>
</tr>
<tr>
<td>C</td>
<td>F</td>
<td>165</td>
<td>37</td>
</tr>
<tr>
<td>D</td>
<td>F</td>
<td>170</td>
<td>37</td>
</tr>
</tbody>
</table>

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)

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Pin</th>
<th>Value</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input loading</td>
<td>a</td>
<td>2.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>b</td>
<td>2.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>c</td>
<td>2.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>d</td>
<td>2.0</td>
<td>Load units</td>
</tr>
<tr>
<td>Drive capability</td>
<td>f</td>
<td>35</td>
<td>Load units</td>
</tr>
</tbody>
</table>
## Comparative view of digital logic technologies

<table>
<thead>
<tr>
<th>Technology</th>
<th>Maximum clock speed</th>
<th>Maximum gate count</th>
<th>Maximum I/Os</th>
</tr>
</thead>
<tbody>
<tr>
<td>GaAs bipolar</td>
<td>100 GHz</td>
<td>500</td>
<td>30</td>
</tr>
<tr>
<td>GaAs fet</td>
<td>30 GHz</td>
<td>10K</td>
<td>300</td>
</tr>
<tr>
<td>Si ECL</td>
<td>10 GHz</td>
<td>10M</td>
<td>500</td>
</tr>
<tr>
<td>Si CMOS</td>
<td>8 GHz</td>
<td>50M</td>
<td>1000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Within Si CMOS</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Full-custom</td>
<td>8 GHz</td>
<td>50M</td>
<td>1000</td>
</tr>
<tr>
<td>Standard cell</td>
<td>4 GHz</td>
<td>25M</td>
<td>1000</td>
</tr>
<tr>
<td>Gate array</td>
<td>2 GHz</td>
<td>5M</td>
<td>1000</td>
</tr>
<tr>
<td>FPGA</td>
<td>175 MHz</td>
<td>500K</td>
<td>800</td>
</tr>
<tr>
<td>CPLD</td>
<td>150 MHz</td>
<td>10K</td>
<td>200</td>
</tr>
<tr>
<td>PAL</td>
<td>200 MHz</td>
<td>500</td>
<td>60</td>
</tr>
</tbody>
</table>
Addition of two integers serially, l.s.b first

\[ \text{let } Y = a \times 5 \text{ in ...} \]

Bit-serial multiplication of an integer by a hardwired constant

\[ \text{let } y = a \times 5 \text{ in ...} \]
Design partitioning: The Cambridge Fast Ring
Design partitioning: An external modem
Design partitioning: A Miniature Radio Module

Multi-chip module or mini PCB

FLASH memory chip

Microcontroller

Hop Controller

Baseband Modem

ADC

DAC

Digital Integrated Circuit

www.bluetooth.org
www.csr.com

Analog (RF) Integrated Circuit

Carrier Oscillator 2.4 GHz

IF Amps

RF Amps

Antenna

Data Interfaces

50
A simple edge detector

Data in -> D flip-flop -> D flip-flop -> Edge Pulse

Clock
A Microcontroller

- Microprocessor (8 bit generally)
- RAM (e.g. 2 K bytes)
- OTP EPROM (e.g. 8 K bytes)
- Clock Osc
- Power Up reset
- Microcontroller
- Counters and Timers
- Programmable IO
- UART
- I/O wires OR external bus
- Serial TX and RX
- Internal A and D busses
- Clock
- Reset capacitor
LEDs wired in a matrix to reduce external pin count
IR Handset Internal Circuit

Scan multiplexed keyboard

Clock capacitor

Single chip containing all semiconductors

Infra-red transmit diodes

Battery
Scan multiplex logic for an LED pixel-mapped display

One col line is logic one at a time.

2^N col lines
Addition of pseudo dual-porting logic

Write address

Write strobe bar

Write data

Broadside tri-state buffer

WE

Pixel RAM

MUX2

N bit COUNTER

N

BINARY to UNARY DECODER

SCAN MULTIPLEXED DISPLAY MATRIX

Row
Use of a ROM as a function look-up table

A to D convertor

Look-up table ROM

D to A convertor

Sample clock 44.1 kHz

65536 by 16 ROM

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

A to D convertor

D to A convertor

Static RAM 65536 by 16 bits

Timing generator circuit

Amplifier

Derived clock, 44.1 kHz

16 bit synchronous counter

16 bit

Read cycle

Write cycle

Clock 88.2

Clock 44.1

Counter Output

N-1

N

N+1

RAM data pins

Old sample replay

New sample write

RAMWE

RAMOE

58
Merge unit block diagram

Bit spacing is reciprocal of 31.25 kbaud, which is 32 microseconds.

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

Midi In 0

Serial to par

Remove status

FIFO Queue

Merged midi output

Midi In 1

Serial to par

Remove status

Queue

Par to serial

Insert running status

Queue

Meger core function

8 24

24

24

8

24

24

24

60
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

Composite Video

NTSC/PAL decoder -> Video Resizer -> 8kx8 SRAM for tile conversion -> 256k video fifo frame buffer

24 bit to 24/16/8 bit RGB/YUV -> C-Cube CL550 JPEG coprocessor

HiFi Audio Codec

Fiber

100 M b/s TAXI interface

80C654 Microcontroller for control / signalling

received ATM cells -> Xilinx 3190 ATM interface control & AAL-5 frame generator

8 -> 2kx8 Dual Port SRAM for assembling whole cells

Xilinx 3190 ATM Cell constructor

16