Processors, FPGAs, and ASICs

Stephen A. Edwards

Columbia University

Spring 2014
Spectrum of IC choices

Flexible, efficient

Full Custom

ASIC

Gate Array

FPGA

PLD

GP Processor

Multifunction

Fixed-function

You choose

Polygons (Intel)

Circuit (Sony)

Wires

Logic network

Logic function

Program (e.g., ARM)

Program (e.g., DSP)

Settings (e.g., Ethernet Ctrl.)

Part number (e.g., 74HCT00)

Cheap, quick to design
An N-Channel MOS Transistor

Gate at 0V: Off

SiO₂

Drain

Source

p (holes)

Ammeter

3 V

0 V
An N-Channel MOS Transistor

Gate positive: On
CMOS Inverter Layout

Cross Section Through N-channel FET

Top View
Two-input NAND gate:
- two n-FETs in series;
- two p-FETs in parallel
The CMOS NAND Gate

Both inputs 0:
Both p-FETs turned on
Output pulled high
The CMOS NAND Gate

One input 1, the other 0:
One p-FET turned on
Output pulled high
One n-FET turned on, but does not control output
The CMOS NAND Gate

Both inputs 1:
- Both n-FETs turned on
- Output pulled low
- Both p-FETs turned off
Full Custom: Intel 4004 Masks
Full Custom: Intel 4004 Die Photograph
Standard Cell ASICs
Standard Cell ASICs
Channeled Gate Arrays
Channeled Gate Arrays
Sea-of-Gates Gate Arrays
FPGAs: Floorplan

The diagram illustrates a typical floorplan of a Field-Programmable Gate Array (FPGA). The layout includes:

- **DLL (Delay Line Logic)**
- **BLOCK RAM**
- **CLBs (Configurable Logic Blocks)**
- **I/O LOGIC**

Each section of the diagram represents a portion of the FPGA, with CLBs and BLOCK RAM blocks arranged in a grid-like pattern. The DLLs are located at the top and bottom edges of the FPGA, providing clock distribution and timing accuracy. The I/O logic is situated at the outer edges, facilitating input and output connections.
FPGAs: Routing

- Single-length line Switch Matrix connections
- Six pass transistors per switch matrix interconnect point
- Double-length lines in CLB array
- CLB
- Switch matrices

Diagram: Routing in FPGA with CLBs and switch matrices.
FPGAs: CLB

Carry and Control Logic

Look-Up Table

G4
G3
G2
G1
F5IN
BY
SR
F4
F3
F2
F1
BX
CIN
CLK
CE
COUT
YB
Y
YQ
XB
X
XQ
D
S
Q
CK
EC
R

Look-Up Table

I4
I3
I2
I1
O

I4
I3
I2
I1
O

I4
I3
I2
I1
O
PLAs/CPLDs: The 22v10

First Fuse Numbers

Increments

Macro-cell

R = 5809
P = 5808

R = 5811
P = 5810

R = 5813
P = 5812

R = 5815
P = 5814

R = 5817
P = 5816

Asynchronous Reset (to all registers)
Example: Euclid’s Algorithm

```c
int gcd(int m, int n) {
    int r;
    while ((r = m % n) != 0) {
        m = n;
        n = r;
    }
    return n;
}
```
**i386 Programmer’s Model**

<table>
<thead>
<tr>
<th>31</th>
<th>0</th>
<th></th>
<th>15</th>
<th>0</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>eax</td>
<td>Mostly General-Purpose Registers</td>
<td>cs</td>
<td>Code segment</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ebx</td>
<td></td>
<td>ds</td>
<td>Data segment</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ecx</td>
<td>Source index</td>
<td>ss</td>
<td>Stack segment</td>
<td></td>
<td></td>
</tr>
<tr>
<td>edx</td>
<td>Destination index</td>
<td>es</td>
<td>Extra segment</td>
<td></td>
<td></td>
</tr>
<tr>
<td>esi</td>
<td></td>
<td>fs</td>
<td>Data segment</td>
<td></td>
<td></td>
</tr>
<tr>
<td>edi</td>
<td>Base pointer</td>
<td>gs</td>
<td>Data segment</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ebp</td>
<td>Stack pointer</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>esp</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>eflags</td>
<td>Status word</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>eip</td>
<td>Instruction Pointer</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Euclid on the i386

```
gcd:  pushl %ebp
    movl %esp,%ebp
    pushl %ebx
    movl 8(%ebp),%eax
    movl 12(%ebp),%ecx
    jmp .L6
  .L4:  movl %ecx,%eax
    movl %ebx,%ecx
  .L6:  cltd
    idivl %ecx
    movl %edx,%ebx
    testl %edx,%edx
    jne .L4
    movl %ecx,%eax
    movl -4(%ebp),%ebx
    leave
    ret
```
SPARC Programmer’s Model

### Global Registers
- r0
- r1
- r7
- r8/o0

### Output Registers
- r14/o6
- r15/o7
- r16/l0
- r23/l7

### Local Registers
- r1

### Always 0
- 31 0

### Input Registers
- r24/i0

### Frame Pointer
- r30/i6
- r31/i7

### Return Address
- PSW
- PC
- nPC

### Status Word
- Program Counter
- Next PC
SPARC Register Windows

The output registers of the calling procedure become the inputs to the called procedure.

The global registers remain unchanged.

The local registers are not visible across procedures.
Euclid on the SPARC

gcd:
    save %sp, -112, %sp
    mov %i0, %o1
    b .LL3
    mov %i1, %i0
    mov %i0, %o1
    b .LL3
    mov %i1, %i0
.LL5:
    mov %o0, %i0
.LL3:
    mov %o1, %o0
    call .rem, 0
    mov %i0, %o1
    cmp %o0, 0
    bne .LL5
    mov %i0, %o1
    ret
    restore
Motorola DSP56301

Peripherals and Features:
- **Triple Timer**
- **Internal Data Bus Switch**
- **Program RAM**: 4096 × 24 (Default)
- **External Data Bus Switch**
- **External Bus Interface and I-Cache Control**
- **Power Management**
- **DMA Unit**:
  - Six Channel DMA Unit
  - Six Channel DMA Unit
- **Host Interface (HI32)**
- **ESSI**
- **SCI**
- **Address Generation Unit**
- **Boot-strap ROM**
- **Internal Data Bus Switch**
- **Address Expansion Area**
  - PIO_EB
- **Peripheral Expansion Area**
  - PM_EB
  - XM_EB
  - YAB
  - YMB
  - PAB
  - DAB
  - DBB
  - DBD
  - PAB
  - DAB
- **24-Bit DSP56300 Core**
  - **24-bit**
  - **MAC**
  - Two 56-bit Accumulators
  - 56-bit Barrel Shifter
- **Program Address Generator**
- **Program Decode Controller**
- **Program Interrupt Controller**
- **Address Expansion Area**
  - PIO_EB
  - PM_EB
  - XM_EB
  - YMB
  - PAB
  - DAB
  - DBB
  - DBD
  - PAB
  - DAB
- **Data ALU**
  - 24 × 24 + 56 → 56-bit
- **External Address Bus Switch**
  - 24
- **External Bus Interface**
  - 14
- **CONTROL**
- **ADDRESS**
- **DATA**
- **DE**
- **EXTAL**
- **XTAL**
- **RESET**
- **PINIT/NMI**
- **24**
DSP 56000 Programmer’s Model

Source Registers

x1 | x0
---|---
y1 | y0

Accumulator

a2 | a1 | a0
---|---|---
b2 | b1 | b0

Address Registers

r7 | n7 | m7
---|---|---
r4 | n4 | m4
---|---|---
r3 | n3 | m3
---|---|---
r0 | n0 | m0
---|---|---

15 0 15 0 15 0

Program Counter

Status Register

Loop Address

Loop Count

PC Stack

SR Stack

Stack pointer
Motorola DSP56301 ALU

X Data Bus
Y Data Bus
P Data Bus

Immediate Field

Bit Field Unit
and Barrel Shifter

Accumulator
Shifter
Immediate Field
48
56
24
24
56
56
56
56

X Data Bus
Y Data Bus
P Data Bus

Multiplier

Pipeline Register

Forwarding Register

Accumulator
and Rounding Unit

A (56)
B (56)

Accumulator
Shifter

56
56
56

56

56

56

56

24
24

56

Shifter/Limiter

24
24
Motorola DSP56301 AGU

Low Address ALU -> Triple Multiplexer -> High Address ALU

Address ALU

Global Data Bus

Program Address Bus

N0 M0
N1 M1
N2 M2
N3 M3

R0 R4
R1 R5
R2 R6
R3 R7

EP

M4 N4
M5 N5
M6 N6
M7 N7

XAB YAB PAB
move  #samples, r0
move  #coeffs, r4
move  #n-1, m0
move  m0, m4
movep y:input, x:(r0)
clr   a       x:(r0)+, x0   y:(r4)+, y0
rep   #n-1
mac   x0,y0,a  x:(r0)+, x0   y:(r4)+, y0
macr  x0,y0,a  (r0)-
movep  a, y:output
FIR in One ’C6 Assembly Instruction

Load a halfword (16 bits)

Do this on unit D1

FIRLOOP:

LDH .D1 *A1++, A2 ; Fetch next sample

LDH .D2 *B1++, B2 ; Fetch next coeff.

SUB .L2 B0, 1, B0 ; Decrement count

[B0] SUB .L2 B0, 1, B0 ; Decrement count

[B0] B .S2 FIRLOOP ; Branch if non-zero

MPY .M1X A2, B2, A3 ; Sample × Coeff.

ADD .L1 A4, A3, A4 ; Accumulate result

Use the cross path

Predicated instruction (only if B0 non-zero)

Run in parallel
AX88796 Ethernet Controller

- **8K* 16 SRAM and Memory Arbiter**
- **SEEPROM I/F**
- **NE2000 Registers**
- **Remote DMA FIFOs**
- **Host Interface**
- **STA**
- **MAC Core & PHY+ Tranceiver**
- **SPP / GPIO**
- **Print Port or General I/O**
- **SMDC**
- **SMDIO**
- **TPI, TPO**
- **MII I/F**
- **Ctl BUS**
- **SA[9:0]**
- **SD[15:0]**
# Ethernet Controller Registers

**PAGE 0 (PS1=0,PS0=0)**

<table>
<thead>
<tr>
<th>OFFSET</th>
<th>READ</th>
<th>WRITE</th>
</tr>
</thead>
<tbody>
<tr>
<td>00H</td>
<td>Command Register (CR)</td>
<td>Command Register (CR)</td>
</tr>
<tr>
<td>01H</td>
<td>Page Start Register (PSTART)</td>
<td>Page Start Register (PSTART)</td>
</tr>
<tr>
<td>02H</td>
<td>Page Stop Register (PSTOP)</td>
<td>Page Stop Register (PSTOP)</td>
</tr>
<tr>
<td>03H</td>
<td>Boundary Pointer (BNRY)</td>
<td>Boundary Pointer (BNRY)</td>
</tr>
<tr>
<td>04H</td>
<td>Transmit Status Register (TSR)</td>
<td>Transmit Page Start Address (TPSR)</td>
</tr>
<tr>
<td>05H</td>
<td>Number of Collisions Register (NCR)</td>
<td>Transmit Byte Count Register 0 (TBCR0)</td>
</tr>
<tr>
<td>06H</td>
<td>Current Page Register (CPR)</td>
<td>Transmit Byte Count Register 1 (TBCR1)</td>
</tr>
<tr>
<td>07H</td>
<td>Interrupt Status Register (ISR)</td>
<td>Interrupt Status Register (ISR)</td>
</tr>
<tr>
<td>08H</td>
<td>Current Remote DMA Address 0 (CRDA0)</td>
<td>Remote Start Address Register 0 (RSAR0)</td>
</tr>
<tr>
<td>09H</td>
<td>Current Remote DMA Address 1 (CRDA1)</td>
<td>Remote Start Address Register 1 (RSAR1)</td>
</tr>
<tr>
<td>0AH</td>
<td>Reserved</td>
<td>Remote Byte Count 0 (RBCR0)</td>
</tr>
<tr>
<td>0BH</td>
<td>Reserved</td>
<td>Remote Byte Count 1 (RBCR1)</td>
</tr>
<tr>
<td>0CH</td>
<td>Receive Status Register (RSR)</td>
<td>Receive Configuration Register (RCR)</td>
</tr>
</tbody>
</table>
Philips SAA7114H Video Decoder

[Diagram of Philips SAA7114H Video Decoder with various connections and labels for different components such as clock generation, input output, I/O control, analog dual ADC, digital decoder, boundary scan test, audio clock generation, real-time output, expansion port pin mapping, FIFOs, event controller, programming register array, and video FIFO.]
<table>
<thead>
<tr>
<th>REGISTER FUNCTION</th>
<th>SUB ADDR (HEX)</th>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chip version: register 00H</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chip version (read only)</td>
<td>00</td>
<td>ID07</td>
<td>ID06</td>
<td>ID05</td>
<td>ID04</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Video decoder: registers 01H to 2FH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Frontend part: registers 01H to 05H</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Horizontal increment delay</td>
<td>01</td>
<td>(I)</td>
<td>(I)</td>
<td>(I)</td>
<td>(I)</td>
<td>IDEL3</td>
<td>IDEL2</td>
<td>IDEL1</td>
<td>IDEL0</td>
</tr>
<tr>
<td>Analog input control 1</td>
<td>02</td>
<td>FUSE1</td>
<td>FUSE0</td>
<td>GUDL1</td>
<td>GUDL0</td>
<td>MODE3</td>
<td>MODE2</td>
<td>MODE1</td>
<td>MODE0</td>
</tr>
<tr>
<td>Analog input control 2</td>
<td>03</td>
<td>(I)</td>
<td>HLNSR</td>
<td>VBSL</td>
<td>WPOFF</td>
<td>HOLDG</td>
<td>GAFIX</td>
<td>GA128</td>
<td>GA118</td>
</tr>
<tr>
<td>Analog input control 3</td>
<td>04</td>
<td>GA17</td>
<td>GA16</td>
<td>GA15</td>
<td>GA14</td>
<td>GA13</td>
<td>GA12</td>
<td>GA11</td>
<td>GA10</td>
</tr>
<tr>
<td>Analog input control 4</td>
<td>05</td>
<td>GA27</td>
<td>GA26</td>
<td>GA25</td>
<td>GA24</td>
<td>GA23</td>
<td>GA22</td>
<td>GA21</td>
<td>GA20</td>
</tr>
<tr>
<td><strong>Decoder part: registers 06H to 2FH</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Horizontal sync start</td>
<td>06</td>
<td>HSB7</td>
<td>HSB6</td>
<td>HSB5</td>
<td>HSB4</td>
<td>HSB3</td>
<td>HSB2</td>
<td>HSB1</td>
<td>HSB0</td>
</tr>
<tr>
<td>Horizontal sync stop</td>
<td>07</td>
<td>HSS7</td>
<td>HSS6</td>
<td>HSS5</td>
<td>HSS4</td>
<td>HSS3</td>
<td>HSS2</td>
<td>HSS1</td>
<td>HSS0</td>
</tr>
<tr>
<td>Sync control</td>
<td>08</td>
<td>AUF D</td>
<td>FSEL</td>
<td>FOET</td>
<td>HTC1</td>
<td>HTC0</td>
<td>HPLL</td>
<td>VNO1</td>
<td>VNO0</td>
</tr>
<tr>
<td>Luminance control</td>
<td>09</td>
<td>BYPS</td>
<td>YCOMB</td>
<td>LDEL</td>
<td>LUBW</td>
<td>LUF1</td>
<td>LUF3</td>
<td>LUF2</td>
<td>LUF0</td>
</tr>
<tr>
<td>Luminance brightness control</td>
<td>0A</td>
<td>DBR17</td>
<td>DBR16</td>
<td>DBR15</td>
<td>DBR14</td>
<td>DBR13</td>
<td>DBR12</td>
<td>DBR11</td>
<td>D BR10</td>
</tr>
<tr>
<td>Luminance contrast control</td>
<td>0B</td>
<td>DCON7</td>
<td>DCON6</td>
<td>DCON5</td>
<td>DCON4</td>
<td>DCON3</td>
<td>DCON2</td>
<td>DCON1</td>
<td>DCON0</td>
</tr>
<tr>
<td>Chrominance saturation control</td>
<td>0C</td>
<td>DSAT7</td>
<td>DSAT6</td>
<td>DSAT5</td>
<td>DSAT4</td>
<td>DSAT3</td>
<td>DSAT2</td>
<td>DSAT1</td>
<td>DSAT0</td>
</tr>
<tr>
<td>Chrominance hue control</td>
<td>0D</td>
<td>HUEC7</td>
<td>HUEC6</td>
<td>HUEC5</td>
<td>HUEC4</td>
<td>HUEC3</td>
<td>HUEC2</td>
<td>HUEC1</td>
<td>HUEC0</td>
</tr>
<tr>
<td>Chrominance control 1</td>
<td>0E</td>
<td>CDTO</td>
<td>CSTD2</td>
<td>CSTD1</td>
<td>CSTD0</td>
<td>DCVF</td>
<td>FCTC</td>
<td>(I)</td>
<td>CCOMB</td>
</tr>
<tr>
<td>Chrominance gain control</td>
<td>0F</td>
<td>OFF-U1</td>
<td>OFF-U0</td>
<td>OFF-V1</td>
<td>OFF-V0</td>
<td>CHB6W</td>
<td>LCB8W</td>
<td>LCB8W</td>
<td>LCB8V</td>
</tr>
<tr>
<td>Chrominance control 2</td>
<td>10</td>
<td>COLO</td>
<td>RTP1</td>
<td>HDEL1</td>
<td>HDEL0</td>
<td>RTP0</td>
<td>YDEL2</td>
<td>YDEL1</td>
<td>YDEL0</td>
</tr>
<tr>
<td>Mode/delay control</td>
<td>11</td>
<td>RTSE13</td>
<td>RTSE12</td>
<td>RTSE11</td>
<td>RTSE10</td>
<td>RTSE9</td>
<td>RTSE8</td>
<td>RTSE7</td>
<td>RTSE6</td>
</tr>
<tr>
<td>RT/X-port output control</td>
<td>12</td>
<td>RTCE</td>
<td>XRH5</td>
<td>XRV5</td>
<td>XRV50</td>
<td>HLSEL</td>
<td>OFTS2</td>
<td>OFTS1</td>
<td>OFTS0</td>
</tr>
<tr>
<td>Analog/ADC/compatibility control</td>
<td>13</td>
<td>CM99</td>
<td>UPTCV</td>
<td>AOSL1</td>
<td>AOSL0</td>
<td>XOUTE</td>
<td>OLD6B</td>
<td>APCK1</td>
<td>APCK0</td>
</tr>
<tr>
<td>VGATE start, FID change</td>
<td>15</td>
<td>VSTA7</td>
<td>VSTA6</td>
<td>VSTA5</td>
<td>VSTA4</td>
<td>VSTA3</td>
<td>VSTA2</td>
<td>VSTA1</td>
<td>VSTA0</td>
</tr>
<tr>
<td>VGATE stop</td>
<td>16</td>
<td>VSTO7</td>
<td>VSTO6</td>
<td>VSTO5</td>
<td>VSTO4</td>
<td>VSTO3</td>
<td>VSTO2</td>
<td>VSTO1</td>
<td>VSTO0</td>
</tr>
<tr>
<td>Miscellaneous/VGATE MSBs</td>
<td>17</td>
<td>LLCE</td>
<td>LLCE2</td>
<td>(I)</td>
<td>(I)</td>
<td>VGPS</td>
<td>VSTO5</td>
<td>VSTA8</td>
<td></td>
</tr>
</tbody>
</table>
Fixed-function: The 7400 series

7400
Quad NAND Gate

74374
Octal D Flip-Flop