Processors, FPGAs, and ASICs

CSEE W4840

Prof. Stephen A. Edwards

Columbia University
Spectrum of IC choices

You choose

Full Custom polygons (Intel)

ASIC circuit (Sony)

Gate Array wires

FPGA logic network

PLD logic function

GP Processor program (e.g., Pentium)

SP Processor program (e.g., DSP)

Multifunction settings (e.g., Ethernet)

Fixed-function part number (e.g., 74LS00)
NMOS Transistor Cross Section
Inverter Transistors and Layout

\[ V_{dd} \]

\[ x \]

\[ V_{ss} \]

\[ \bar{x} \]

\[ V_{dd} \]

\[ \bar{x} \]

\[ V_{ss} \]

\[ x \]
NAND Gate Transistors and Layout

The diagram illustrates a NAND gate with the following inputs and outputs:

- **Input**: $x$ and $y$
- **Output**: $\overline{x \land y}$

The gate is shown with voltage levels $V_{dd}$ and $V_{ss}$.
Full-custom ICs
Standard Cell ASICs
Standard Cell ASICs
Channeled Gate Arrays

[Diagram of a channeled gate array with labels for Underpasses, NFETs PFETs, Gates, Ground/Power, Vss, Vdd, Bonding Pad, Big N, Big P, Protection Diode]
Channeled Gate Arrays
Sea-of-Gates Gate Arrays
Architectural Description

Spartan-IIE Array

The Spartan-IIE user-programmable gate array, shown in Figure 1, is composed of five major configurable elements:

- IOBs provide the interface between the package pins and the internal logic
- CLBs provide the functional elements for constructing most logic
- Dedicated block RAM memories of 4096 bits each
- Clock DLLs for clock-distribution delay compensation and clock domain control
- Versatile multi-level interconnect structure

As can be seen in Figure 1, the CLBs form the central logic structure with easy access to all support and routing structures. The IOBs are located around all the logic and memory elements for easy and quick routing of signals on and off the chip.

Values stored in static memory cells control all the configurable logic elements and interconnect resources. These values load into the memory cells on power-up, and can reload if necessary to change the function of the device. Each of these elements will be discussed in detail in the following sections.

Input/Output Block

The Spartan-IIE IOB, as seen in Figure 2, features inputs and outputs that support a wide variety of I/O signaling standards. These high-speed inputs and outputs are capable of supporting various state of the art memory and bus interfaces.

Table 1 lists several of the standards which are supported along with the required reference, output, and termination voltages needed to meet the standard.

The three IOB registers function either as edge-triggered D-type flip-flops or as level-sensitive latches. Each IOB has a clock signal (CLK) shared by the three registers and independent Clock Enable (CE) signals for each register.
FPGAs: Routing

Single-length line Switch Matrix connections

Six pass transistors per switch matrix interconnect point

Double-length lines in CLB array
The F5 multiplexer in each slice combines the function generator outputs (Figure 5). This combination provides either a function generator that can implement any 5-input function, a 4:1 multiplexer, or selected functions of up to nine inputs.

Similarly, the F6 multiplexer combines the outputs of all four function generators in the CLB by selecting one of the two F5-multiplexer outputs. This permits the implementation of any 6-input function, an 8:1 multiplexer, or selected functions of up to 19 inputs.
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>15</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>eax</td>
<td>Mostly</td>
<td>cs</td>
<td>Code segment</td>
</tr>
<tr>
<td>ebx</td>
<td>General-Purpose-Registers</td>
<td>ds</td>
<td>Data segment</td>
</tr>
<tr>
<td>ecx</td>
<td></td>
<td>ss</td>
<td>Stack segment</td>
</tr>
<tr>
<td>edx</td>
<td></td>
<td>es</td>
<td>Extra segment</td>
</tr>
<tr>
<td>esi</td>
<td>Source index</td>
<td>fs</td>
<td>Data segment</td>
</tr>
<tr>
<td>edi</td>
<td>Destination index</td>
<td>gs</td>
<td>Data segment</td>
</tr>
<tr>
<td>ebp</td>
<td>Base pointer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>esp</td>
<td>Stack pointer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>eflags</td>
<td>Status word</td>
<td></td>
<td></td>
</tr>
<tr>
<td>eip</td>
<td>Instruction Pointer</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
<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>r0</td>
<td>Always 0</td>
</tr>
<tr>
<td>r1</td>
<td>Global Registers</td>
</tr>
<tr>
<td>r7</td>
<td>Output Registers</td>
</tr>
<tr>
<td>r8/o0</td>
<td></td>
</tr>
<tr>
<td>r14/o6</td>
<td>Stack Pointer</td>
</tr>
<tr>
<td>r15/o7</td>
<td>Local Registers</td>
</tr>
<tr>
<td>r16/l0</td>
<td></td>
</tr>
<tr>
<td>r17/l7</td>
<td></td>
</tr>
<tr>
<td>r24/i0</td>
<td>Input Registers</td>
</tr>
<tr>
<td>r30/i6</td>
<td>Frame Pointer</td>
</tr>
<tr>
<td>r31/i7</td>
<td>Return Address</td>
</tr>
<tr>
<td>PSW</td>
<td>Status Word</td>
</tr>
<tr>
<td>PC</td>
<td>Program Counter</td>
</tr>
<tr>
<td>nPC</td>
<td>Next PC</td>
</tr>
</tbody>
</table>

SPARC Programmer’s Model
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.
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
The block diagram in Figure 1-1 illustrates these buses among other components.

1.6 DMA

The DMA block has the following features:

- Six DMA channels supporting internal and external accesses
- One-, two-, and three-dimensional transfers (including circular buffering)
- End-of-block-transfer interrupts
- Triggering from interrupt lines and all peripherals

Figure 1-1. DSP56301 Block Diagram
DSP 56000 Programmer’s Model

Source Registers

Accumulator

Program Counter

Status Register

Loop Address

Loop Count

PC Stack

SR Stack

Stack pointer

55 4847 2423 0

x1 x0

y1 y0

a2 a1 a0

b2 b1 b0

15 0

r7 n7 m7

r4 n4 m4

r3 n3 m3

r0 n0 m0
The Data ALU registers can be read or written over the X Data Bus (XDB) and the Y Data Bus (YDB) as 24- or 48-bit operands. The source operands for the Data ALU, which can be 24, 48, or 56 bits, always originate from Data ALU registers. The results of all Data ALU operations are stored in an accumulator. The Data ALU runs in 16-bit Arithmetic mode when the SA bit in the Status Register (SR) is set. For details on the SR, see Chapter 5, Program Control Unit.
The offset \( N \) (stored in the respective offset register)

- Minus \( N \) to the selected address register

The offset adder and the reverse-carry adder operate in parallel and share common inputs. The only difference between them is that the carry propagates in opposite directions. Test logic determines which of the three summed results of the full adders is output.

Figure 4-1 shows a block diagram of the AGU.

Each Address ALU can update one address register from its respective address register file during one instruction cycle. The contents of the associated modifier register specify the type of arithmetic to be used in the address register update calculation. The modifier value is decoded in the Address ALU.

The two Address ALUs can generate up to two addresses every instruction cycle:

- One for the PAB,
- One for the XAB,
- One for the YAB,
- One for the XAB and one for the YAB

The AGU can directly address 16,777,216 locations on each of the XAB, YAB, and PAB. Using a register triplet to address each operand, the two independent ALUs can work with the two data memories to feed two operands to the Data ALU in a single cycle.
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
Figure 2±1. TMS320C62x CPU Data Paths

Data path A

- Data path B

Register file A (A0–A15)

Register file B (B0–B15)

Control register file

Processors, FPGAs, and ASICs – p. 27/?
FIR in One ’C6 Assembly Instruction

Load a halfword (16 bits)
Do this on unit D1

FIRLOOP:

1. LDH .D1 *A1++, A2 ; Fetch next sample
2. LDH .D2 *B1++, B2 ; Fetch next coeff.
3. [B0] SUB .L2 B0, 1, B0 ; Decrement count
4. [B0] B .S2 FIRLOOP ; Branch if non-zero
5. MPY .M1X A2, B2, A3 ; Sample × Coeff.
6. ADD .L1 A4, A3, A4 ; Accumulate result

Use the cross path
Predicated instruction (only if B0 non-zero)
Run these instruction in parallel
AX88796 Fast Ethernet Controller is a high performance and highly integrated local CPU bus Ethernet Controller with embedded 10/100Mbps PHY/Transceiver and 8K*16 bit SRAM. The AX88796 supports both 8 bit and 16 bit local CPU interfaces including MCS-51 series, 80186 series, MC68K series CPU, and ISA bus.

The AX88796 implements both 10Mbps and 100Mbps Ethernet function based on IEEE802.3 / IEEE802.3u LAN standard. The AX88796 also provides an extra IEEE802.3u compliant media-independent interface (MII) to support other media applications. Using MII interface, Home LAN PHY type media can be supported.

As well as, the chip also provides optional Standard Print Port (parallel port interface), can be used for printer server device or treated as simple general I/O port. The chip also supports up to 3/1 additional General Purpose In/Out pins.

The main difference between AX88796 and AX88195 are:
1) Embedded packet buffer memory
2) Built-in 10/100Mbps PHY/Transceiver
3) Replace memory I/F with PHY/Transceiver I/F.
4) Canceling SAX address decoding.
5) Fixing interrupt status can't always clean up problems of AX88195.
6) Adding up to 3/1 general-purpose In/Out pins.

AX88796 uses a 128-pin LQFP low profile package, operates at 25MHz, and has single 3.3V operation with 5V I/O tolerance.

The ultra-low power consumption is an outstanding feature and enlarges the application field. It is suitable for some power-consumption-sensitive products like small size embedded products, PDA (Personal Digital Assistant) and Palm size computer... etc.

1.2 AX88796 Block Diagram:

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

Processors, FPGAs, and ASICs – p. 29/7
# 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]
<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>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>Chip version (read only)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></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>FRONT-END PART: REGISTERS 01H to 05H</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>H1</td>
<td>H0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Analog input control 1</td>
<td>02</td>
<td>FUSE1</td>
<td>FUSE0</td>
<td>GUID1</td>
<td>GUID0</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>HINR</td>
<td>VBSL</td>
<td>WPOFF</td>
<td>HOLDG</td>
<td>GAFIX</td>
<td>GAI28</td>
<td>GAI18</td>
<td>GAI10</td>
</tr>
<tr>
<td>Analog input control 3</td>
<td>04</td>
<td>GAI17</td>
<td>GAI16</td>
<td>GAI15</td>
<td>GAI14</td>
<td>GAI13</td>
<td>GAI12</td>
<td>GAI11</td>
<td>GAI10</td>
</tr>
<tr>
<td>Analog input control 4</td>
<td>05</td>
<td>GAI27</td>
<td>GAI26</td>
<td>GAI25</td>
<td>GAI24</td>
<td>GAI23</td>
<td>GAI22</td>
<td>GAI21</td>
<td>GAI20</td>
</tr>
<tr>
<td>DECODER PART: REGISTERS 06H to 2FH</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>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>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>SUTD</td>
<td>SEL</td>
<td>FOLD</td>
<td>FTC1</td>
<td>FTC0</td>
<td>FTC1</td>
<td>FTC2</td>
<td>FTC3</td>
</tr>
<tr>
<td>Luminance control</td>
<td>09</td>
<td>BYPS</td>
<td>YCOMB</td>
<td>LEDL</td>
<td>LUBW</td>
<td>LUF1</td>
<td>LUF2</td>
<td>LUF3</td>
<td>LUF4</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>DBR10</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>CIDT0</td>
<td>CSTD2</td>
<td>CSTD1</td>
<td>CSTD0</td>
<td>DCFV</td>
<td>FCTC</td>
<td>FCOH</td>
<td>FCOH</td>
</tr>
<tr>
<td>Chrominance gain control</td>
<td>0F</td>
<td>ACGC</td>
<td>CAGN6</td>
<td>CAGN5</td>
<td>CAGN4</td>
<td>CAGN3</td>
<td>CAGN2</td>
<td>CAGN1</td>
<td>CAGN0</td>
</tr>
<tr>
<td>Chrominance control 2</td>
<td>10</td>
<td>OFFU1</td>
<td>OFFU0</td>
<td>OFFV1</td>
<td>OFFV0</td>
<td>CHBW</td>
<td>LCBW2</td>
<td>LCBW1</td>
<td>LCBW0</td>
</tr>
<tr>
<td>Model/delay control</td>
<td>11</td>
<td>COL0</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>RT signal control</td>
<td>12</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>13</td>
<td>RTCE</td>
<td>XFRS</td>
<td>XFRS1</td>
<td>XFRS0</td>
<td>HLSL</td>
<td>OTFS1</td>
<td>OTFS2</td>
<td>OTFS3</td>
</tr>
<tr>
<td>Analog/ADC/compatibility control</td>
<td>14</td>
<td>CM99</td>
<td>UPTCV</td>
<td>AOSLL</td>
<td>AOSLO</td>
<td>XTSC4</td>
<td>OLDSB</td>
<td>APCK1</td>
<td>APCK0</td>
</tr>
<tr>
<td>VGATE start, FLU 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>LLCE2E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Processors, FPGAs, and ASICs – p. 327
Fixed-function: The 7400 series

7400
Quad NAND Gate

74374
Octal D Flip-Flop