Spectrum of IC choices

- **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)

You choose

Polygons: Intel
Circuit: Sony
Wires: Part number: 74LS00
Logic network: FPGA
Logic function: PLD
Program: GP processor
Program: SP processor
Settings: Multifunction
Part number: Fixed-function

NMOS Transistor Cross Section

Inverter Transistors and Layout

NAND Gate Transistors and Layout

Full-custom ICs

Standard Cell ASICs

Channeled Gate Arrays
**Channeled Gate Arrays**

**Sea-of-Gates Gate Arrays**

**FPGAs: Floorplan**

**FPGAs: Routing**

**FPGAs: CLB**

**PLAs/CPLDs: The 22v10**

**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**

```c
    31  0
eax  Mostly cs
ebx  General cs
ecx  Purpose es
edx  Registers ds
esi  Source index fs
edi  Destination index gs
ebp  Base pointer ss
esp  Stack pointer
esp  Instruction Pointer

gcd: pushl %ebp
     movl %ebp,%esp
     pushl %ebx
     movl %ebp,%ecx
     movl $12,%ecx
     jmp .L6
     .L4: movl %ecx,%eax
     movl %ebp,%ecx
     .L6: cltd
     idivl %ecx
     movl %edx,%ebx
     testl %edx,%edx
     jne .L4
     movl %ecx,%eax
     movl $-4,%ebp
     leave
     ret
```

**Euclid on the i386**
Euclid on the SPARC

gcd:
  save %sp, -112, %sp
  mov %10, %10
  b .LL3
  mov %11, %10
  mov %10, %10
  b .LL3
  mov %11, %10
  .LL5:
  mov %00, %10
  .LL3:
  mov %10, %00
  call .rem, 0
  mov %10, %10
  cmp %00, 0
  bne .LL5
  mov %10, %10
  ret
  restore

Motorola DSP56301

Motorola DSP56301 ALU

DSP 56000 Programmer’s Model

Motorola DSP56301 AGU

FIR Filter in 56000

TI TMS320C6000 VLIW DSP

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

1. **Load a halfword (16 bits)**
   - Do this on unit D1

2. **FIRLOOP:**
   - `LDH .D1 *A1++, A2`: Fetch next sample
   - `[B0] SUB .L2 B0, 1, B0`: Decrement count
   - `[B0] B .S2 FIRLOOP`: Branch if non-zero
   - `ADD .L1 A4, A3, A4`: Accumulate result

3. **Use the cross path**
   - Predicated instruction (only if B0 non-zero)

4. **Run these instruction in parallel**

---

**AX88796 Ethernet Controller**

**Ethernet Controller Registers**

<table>
<thead>
<tr>
<th>PAGE 0 (PS1=0,PS0=0)</th>
<th>READ</th>
<th>WRITE</th>
</tr>
</thead>
<tbody>
<tr>
<td>000H Command Register</td>
<td>Command Register CR</td>
<td>Page Start Register PSTART</td>
</tr>
<tr>
<td>010H Page Start Register</td>
<td>Page Start Register PSTART</td>
<td>Page Stop Register PSTOP</td>
</tr>
<tr>
<td>050H Boundary Pointer</td>
<td>Boundary Pointer BRY</td>
<td>Transmit Status Register TSM</td>
</tr>
<tr>
<td>0F0H Transmit Status Register</td>
<td>Transmit Status Register TSM</td>
<td>Transmit Status Register TSM</td>
</tr>
<tr>
<td>0F2H Transmit Page Start Address</td>
<td>Transmit Page Start Address TPSA</td>
<td>Transmit Byte Count Register 0 TBCR0</td>
</tr>
<tr>
<td>0F4H Transmit Page Register</td>
<td>Transmit Page Register TPR</td>
<td>Transmit Byte Count Register 1 TBCR1</td>
</tr>
<tr>
<td>0F6H Transmit Status Register</td>
<td>Transmit Status Register TSM</td>
<td>Transmit Status Register TSM</td>
</tr>
<tr>
<td>0F8H Receive Status Register</td>
<td>Receive Status Register RSR</td>
<td>Remote Page Register RPR</td>
</tr>
<tr>
<td>0FCH Receive Status Register</td>
<td>Remote Page Register RPR</td>
<td>Remote Start Address Register 0 RSAR0</td>
</tr>
<tr>
<td>0FEH Remote Page Start Address</td>
<td>Remote Start Address Register 0 RSAR0</td>
<td>Remote Page Start Address RPR</td>
</tr>
<tr>
<td>0F2H Remote Page Start Address</td>
<td>Remote Page Start Address RPR</td>
<td>Remote Start Address Register 1 RSAR1</td>
</tr>
<tr>
<td>0F4H Transmit Start Address Register 0</td>
<td>Transmit Start Address Register 0 TPSA</td>
<td>Transmit Start Address Register 1 TPSA</td>
</tr>
<tr>
<td>0F6H Remote Start Address Register 0</td>
<td>Remote Start Address Register 0 RSAR0</td>
<td>Remote Start Address Register 1 RSAR1</td>
</tr>
<tr>
<td>0FCH Remote Start Address Register 0</td>
<td>Remote Start Address Register 0 RSAR0</td>
<td>Remote Start Address Register 1 RSAR1</td>
</tr>
<tr>
<td>0FEH Receive Start Address Register 0</td>
<td>Remote Start Address Register 0 RSAR0</td>
<td>Remote Start Address Register 1 RSAR1</td>
</tr>
</tbody>
</table>

---

**Philips SAA7114H Video Decoder**

**SAA7114H Registers, page 1 of 7 (!)**

**Fixed-function: The 7400 series**

- **7400**
  - Quad NAND Gate
- **74374**
  - Octal D Flip-Flop