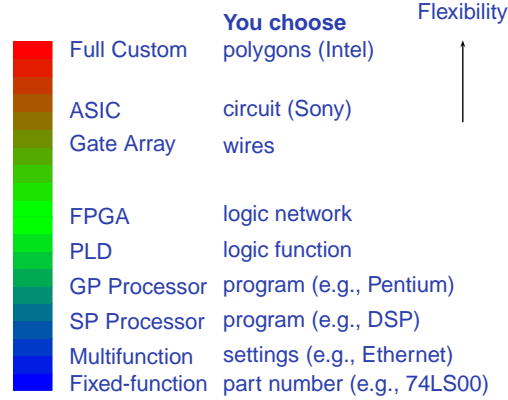


# Processors, FPGAs, and ASICs

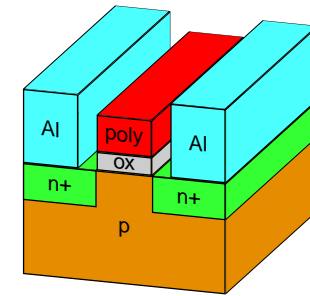
Prof. Stephen A. Edwards  
sedwards@cs.columbia.edu

Columbia University  
Spring 2006

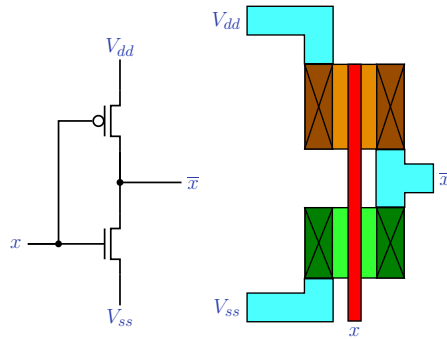
## Spectrum of IC choices



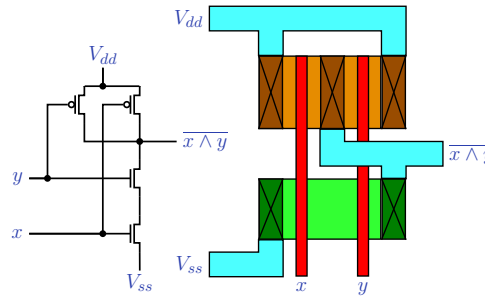
## NMOS Transistor Cross Section



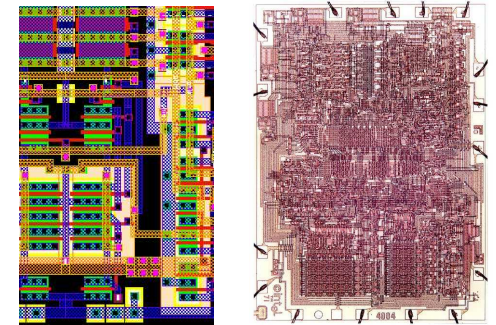
## Inverter Transistors and Layout



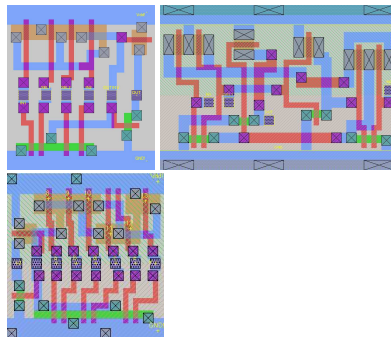
## NAND Gate Transistors and Layout



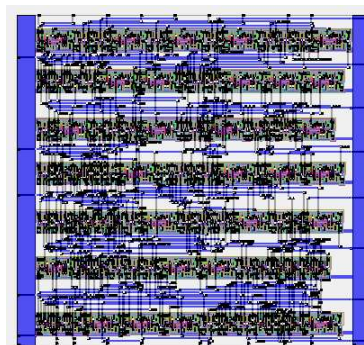
## Full-custom ICs



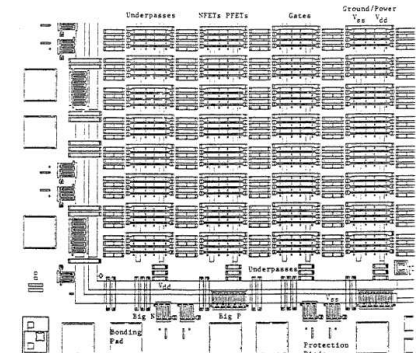
## Standard Cell ASICs



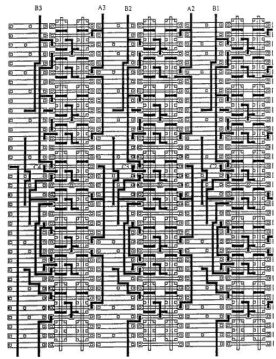
## Standard Cell ASICs



## Channeled Gate Arrays

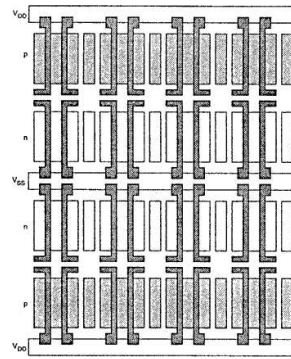


## Channeled Gate Arrays



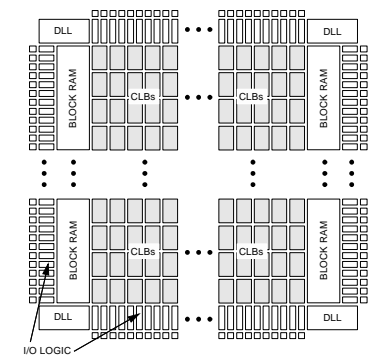
Processors, FPGAs, and ASICs - p. 103

## Sea-of-Gates Gate Arrays



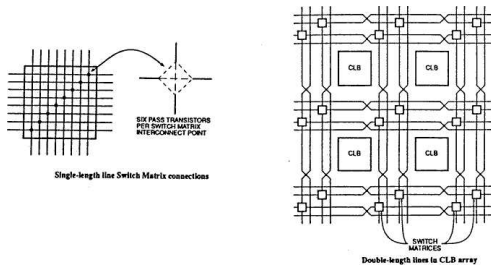
Processors, FPGAs, and ASICs - p. 113

## FPGAs: Floorplan



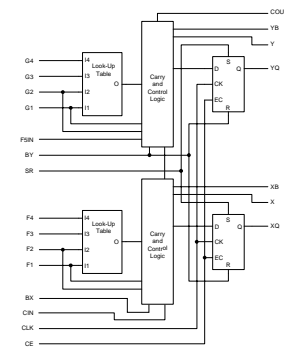
Processors, FPGAs, and ASICs - p. 123

## FPGAs: Routing



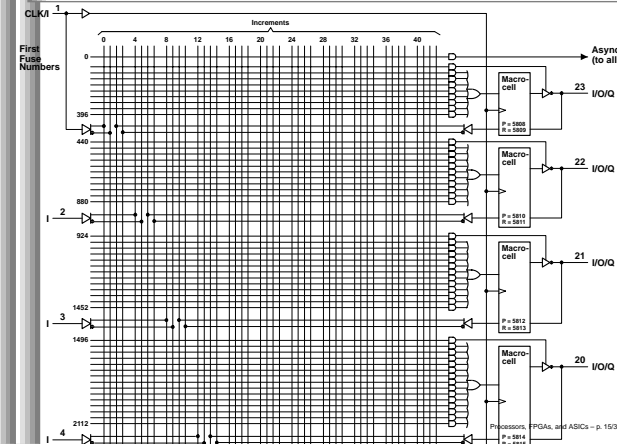
Processors, FPGAs, and ASICs - p. 133

## FPGAs: CLB



Processors, FPGAs, and ASICs - p. 143

## PLAs/CPLDs: The 22v10



Processors, FPGAs, and ASICs - p. 153

## Example: Euclid's Algorithm

```
int gcd(int m, int n)
{
    int r;
    while ((r = m % n) != 0) {
        m = n;
        n = r;
    }
    return n;
}
```

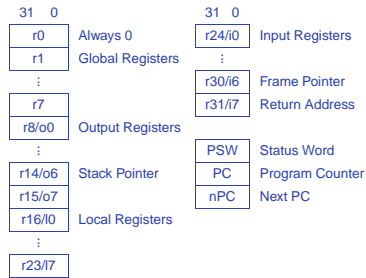
## i386 Programmer's Model

31	0	15	0
eax	Mostly	cs	Code segment
ebx	General-	ds	Data segment
ecx	Purpose-	ss	Stack segment
edx	Registers	es	Extra segment
esi	Source index	fs	Data segment
edi	Destination index	gs	Data segment
ebp	Base pointer		
esp	Stack pointer		
eflags	Status word		
eip	Instruction Pointer		

## Euclid on the i386

```
god:  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:  cld
      idivl %ecx
      movl  %edx, %ebx
      testl %edx, %edx
      jne  .L4
      movl  %ecx, %eax
      movl  -4(%ebp), %ebx
      leave
      ret
```

## SPARC Programmer's Model



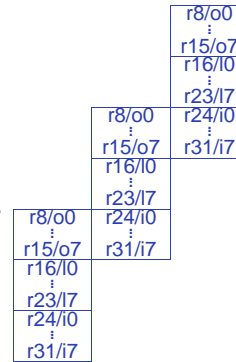
Processors, FPGAs, and ASICs - p. 193

## 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



Processors, FPGAs, and ASICs - p. 203

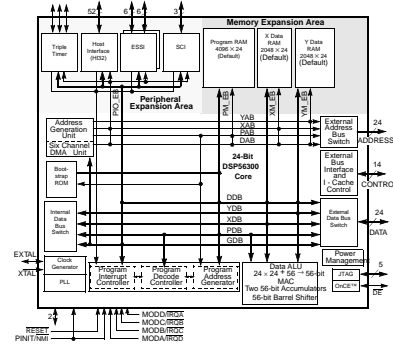
## Euclid on the SPARC

```

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
    
```

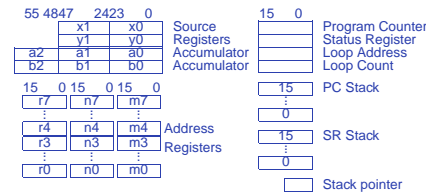
Processors, FPGAs, and ASICs - p. 213

## Motorola DSP56301



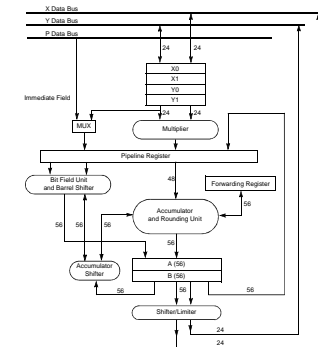
Processors, FPGAs, and ASICs - p. 223

## DSP 56000 Programmer's Model



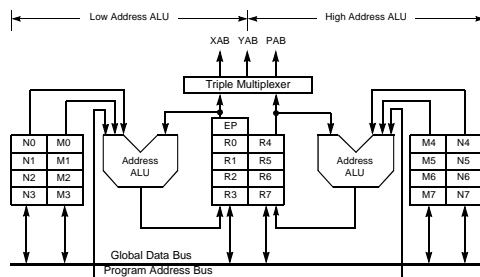
Processors, FPGAs, and ASICs - p. 233

## Motorola DSP56301 ALU



Processors, FPGAs, and ASICs - p. 243

## Motorola DSP56301 AGU



## FIR Filter in 56000

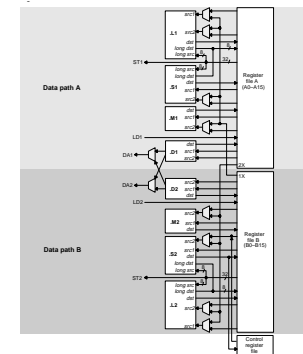
```

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
    
```

## TI TMS320C6000 VLIW DSP



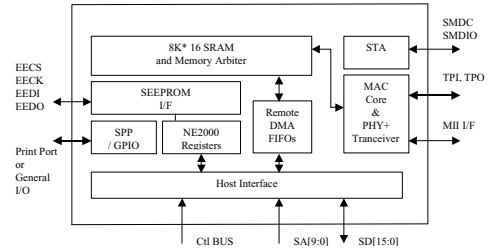
# FIR in One 'C6 Assembly Instruction

```

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

Load a halfword (16 bits)  
 Do this on unit D1  
 Use the cross path  
 Predicated instruction (only if B0 non-zero)  
 Run these instructions in parallel

# AX88796 Ethernet Controller

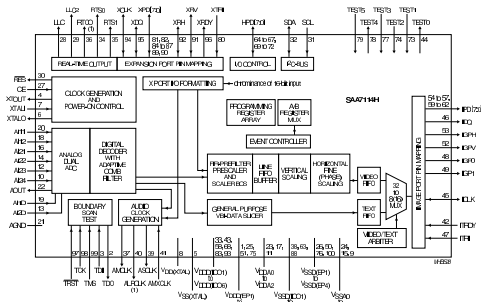


# Ethernet Controller Registers

PAGE 0 (PS1=0, PS0=0)

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

# Philips SAA7114H Video Decoder



# SAA7114H Registers, page 1 of 7 (!)

REGISTER/FUNCTION	SUB ADDR (HEX)	D7	D6	D5	D4	D3	D2	D1	D0
Chip version register 00H	00	1007	1035	1005	1004	-	-	-	-
Chip version (read only)									
Video decoder registers 01H to 0FH									
Decoder input registers 0AH to 0FH									
Horizontal sync delay	01	00	00	00	00	DEE5	DEE2	DEE1	DEE0
Video input control 1	02	FE61	FE60	GE14	GE10	ME05	ME02	ME01	ME00
Video input control 2	03	00	HEF5	VE5L	WPOFF	HE10	GA7K	GA6B	GA6B
Video input control 3	04	GA17	GA16	GA15	GA14	GA13	GA12	GA11	GA10
Video input control 4	05	GA27	GA26	GA25	GA24	GA23	GA22	GA21	GA20
Decoder input registers 0AH to 0FH									
Horizontal sync start	06	HSE7	HSE6	HSE5	HSE4	HSE3	HSE2	HSE1	HSE0
Horizontal sync stop	07	HSE7	HSE6	HSE5	HSE4	HSE3	HSE2	HSE1	HSE0
Sync control	08	AJF0	FSE1	FE11	HE10	HE11	HE11	WCM1	WCM0
Luminance control	09	DE9F	YOCAB	LEL1	LEW0	LEF0	LEF0	LEF1	LEF0
Luminance brightness control	0A	DE9F	DE96	DE95	DE94	DE93	DE92	DE91	DE90
Luminance contrast control	0B	DC0V	DC06	DC05	DC04	DC03	DC02	DC01	DC00
Chrominance saturation control	0C	DS07	DS06	DS05	DS04	DS03	DS02	DS01	DS00
Chrominance hue control	0D	HLE27	HLE26	HLE25	HLE24	HLE23	HLE22	HLE21	HLE20
Chrominance control 1	0E	CD10	CS1E	CS1D	CS1D	DC1F	FC1C	00	CC0AB
Chrominance gain control	0F	AC0C	CS0AB	CS0AB	CS0AB	CS0AB	CS0AB	CS0AB	CS0AB
Chrominance control 2	10	OFFU1	OFFU0	OFFV1	OFFV0	CH0V	LC0AB	LC0A1	LC0A0
Mode select control	11	COLO	RIP1	HE11	HE10	RIT0	YDE2	YDE1	YDE0
Filter select control	12	RISE15	RISE12	RISE11	RISE10	RISE09	RISE08	RISE07	RISE06
Filter port select control	13	RICE	YR05	YR05	YR05	HSEL	CF150	CF151	CF150
Analogue compatibility control	14	CM0	UP1CV	AC0L1	XT0URE	CL050	AP0K1	AP0K0	
Vertical sync change	15	VSR7	VSR6	VSR5	VSR4	VSR3	VSR2	VSR1	VSR0
VGA test	16	VST07	VST06	VST05	VST04	VST03	VST02	VST01	VST00
Microprocessor GATE MEMs	17	LICE	LLCE	00	00	00	VGPS	VST00	VSR0

# Fixed-function: The 7400 series

