Digital Signal Processors (DSP)

- Specialization for scalar product (and FFT)
- Single-cycle processing (memory throughput):
  - parallelism (pipelining)
  - Harvard architecture (program (P), data (X), data (Y) memories)
- Desing for embedding – I/O & host interfaces etc.
- Special arithmetic modes: rounding, saturation
- Special addressing modes: circular buffer, bit-reversal, matrix address

Manufacturers

- Texas Instruments (TMS family)
  - From simple to multiporcessor
  - fixed point and floating point
- Analog Devices (ADSP family, codename “SHARC” etc.)
- Motorola ⟶ now Freescale Semiconductors (DSP56K family)

Also “DSP cores” - VHDL or silicon IP blocks to be embedded into VLSIs.

DSP56002

- Arithmetics:
  - 24-bit (144 dB), fixed-point fraction (-1.0 till +1.0)
  - 2x24+8 bits in accumulators
  - rounding and saturation
- Addressing
  - indirect: `X:(Rn)+Nn` (post-modification)
  - indexed: `X:(Rn+Nn)`
  - absolute: `X:7`
  - immediate: `#0.125`
  - modulus register: M=-1 (no modif) M=0 (bit-reversal) M=other (circ.buffer)
- Memory: 256(X)+256(Y), extendable to 64k

Assembly programming of DSP56002

registers	symbols		
X0, X1, Y0, Y1 A, B	x, y a, b	s	g, h
R0, ..., R7 N0, ..., N7	r n	i	

abs, asl, asr, clr, neg, rnd: `abs a;`
add, sub: `add s, a;`
mpy, mpyr, mac, macr: `mpy ±x,y,a;`
nop

```
move x:ea,g; from memory
move g,x:ea; to memory
move ea; (update Rn)
move g,h;
move #c,g;
```

asembly		meaning		mode
ea	X&Y	ea	R update	
(r)-n			r=r-n;	
(r)+n	(yes)		r=r+n;	
(r)-	(yes)		r=r-1;	
(r)+	(yes)		r=r+1;	
(r)	(yes)	r		indirect
(r+n)		r+n		indexed
c		c		absolute

```
macr -x0,x0,a   a,x:(r3)-   y:(r5)+n5,x0
```

Example – FIR filter

```
N       equ  8
        org  x:0
samples ds  N
        org  y:0
coeffs  dc  0.0286,0.0716,0.1683,0.2458
        org  p:64    ; start address
        init      ;
        move  #samples,r0    ;
        move  #coeffs,r4    ;
        move  #N-1,m0    ;
        move  m0,m4    ;
        repeat      ;
        in   x:(r0)
        clr   a   x:(r0)+,x0    y:(r4)+,y0    ; x(n), h(0)
        .loop   #N-1
        mac  x0,y0,a       x:(r0)+,x0    y:(r4)+,y0    ; x(n-k), h(k)
        .endl
        macr   x0,y0,a   (r0)-       ;
        out   a
        forever     ;
```