

Low-Power DSP's for Wireless Communications

Ingrid Verbauwhede

Department of Electrical Engineering
University of California Los Angeles

ingrid@ee.ucla.edu

David Garrett

Bell Laboratories Australia
Lucent Technologies

chrisn@lucent.com, garrettd@lucent.com

1

ISLPED 2000, Tutorial. Ingrid Verbauwhede, Dave Garrett

Design for low power

Multiple levels of abstraction:

- technology level: dual V_t ,
- circuit level: complementary static CMOS
- logic level: gated clocks
- architectural level: datapath, memory optimization
- system level: idle modes, variable supply voltage

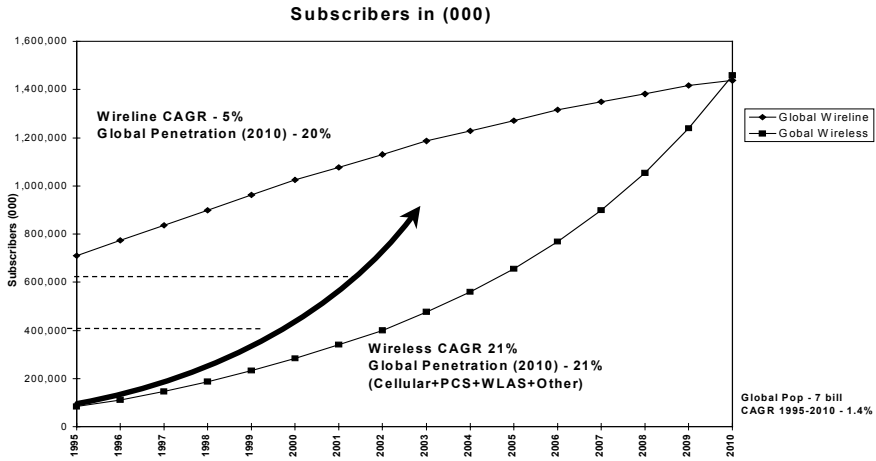
Focus of this presentation: Architectural level

TUNE processor components to the application domain
HENCE reduce overhead, unnecessary computations, unnecessary switching

2

ISLPED 2000, Tutorial. Ingrid Verbauwhede, Dave Garrett

Application domain: Wireless



World-wide deployment of mobile communications is exceeding expectations

3

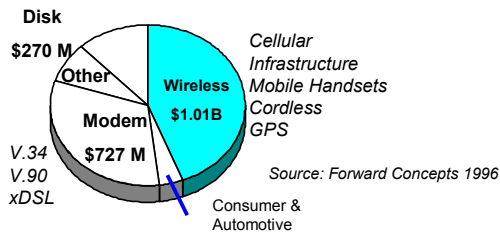
ISLPED 2000, Tutorial. Ingrid Verbauehede, Dave Garrett

Digital Signal Processor Market

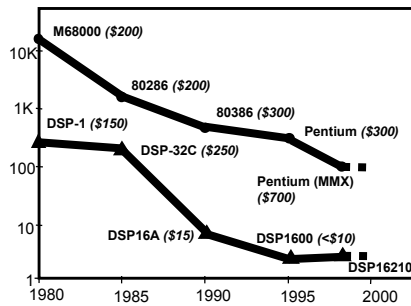
DSP Market

1996: \$2B market, 30% growth rate
1999: \$4.4B
2004: \$19B

Source: Forward Concepts 2000



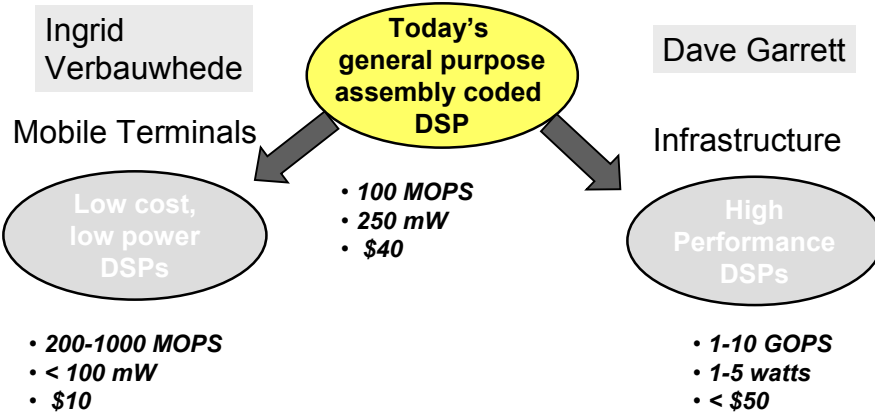
Power (mw/MIP)



4

ISLPED 2000, Tutorial. Ingrid Verbauehede, Dave Garrett

The DSP Market Splits - and so does this tutorial



5

ISLPED 2000, Tutorial. Ingrid Verbauwheide, Dave Garrett

Overview:

- **Introduction**
- **Low Power DSP Architectures for Handsets**
 - Domain Specific Processors
 - DSP Processor Fundamentals
 - Datapath Design, Instruction Set Design
 - Pipeline Control, Memory Architecture, Low Power Design for FIR - Viterbi - Turbo coding
- **High performance DSP Processors for BTS**
 - **2G and 3G Wireless Standards**
 - **Mobile Wireless Basestation Systems**
 - Requirements
 - Convolutional and Turbo coding
 - **High Performance DSP Architectures for 3G Wireless**
 - LU DSP16210, TI 'C6x, Starcore SC140
 - Future Trends - MIMD DSP

6

ISLPED 2000, Tutorial. Ingrid Verbauwheide, Dave Garrett

Domain Specific Processors

Domain specific processors: to combine

- High performance
- Low Power
- High degree of programmability

Application domains that need it:

- Wireless communications (baseband processing)
- Video processors
- Embedded micro controllers
- Etc.

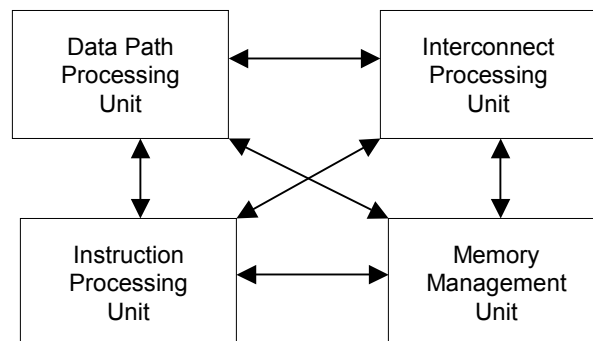
Application domain is narrower, hence need high volume to compensate development cost.

7

ISLPED 2000, Tutorial. Ingrid Verbauwhede, Dave Garrett

DSP Processor Fundamentals

Processor Components [Skillikorn-88]



Adapt **ALL** components to the application domain!

8

ISLPED 2000, Tutorial. Ingrid Verbauwhede, Dave Garrett

Compute intensive functions: evolution of DSP's

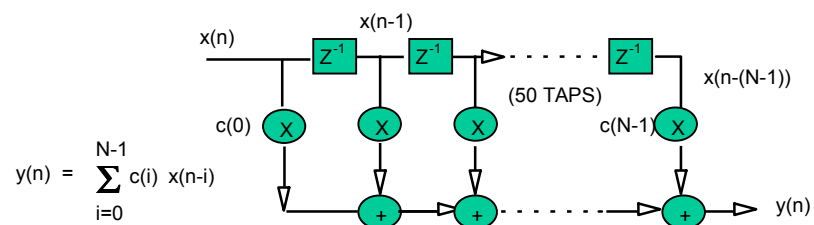
Illustrate with examples used in wireless communications.

- Simple FIR example
- Speed-up of FIR example
- Viterbi acceleration
- Square distance
- Turbo coding

9

ISLPED 2000, Tutorial. Ingrid Verbauwhede, Dave Garrett

FIR implementation



$$\begin{aligned}
 y(0) &= c(0)x(0) + c(1)x(-1) + c(2)x(-2) + \dots + c(N-1)x(1-N); \\
 y(1) &= c(0)x(1) + c(1)x(0) + c(2)x(-1) + \dots + c(N-1)x(2-N); \\
 y(2) &= c(0)x(2) + c(1)x(1) + c(2)x(0) + \dots + c(N-1)x(3-N); \\
 &\dots \\
 y(n) &= c(0)x(n) + c(1)x(n-1) + c(2)x(n-2) + \dots + c(N-1)x(n-(N-1));
 \end{aligned}$$

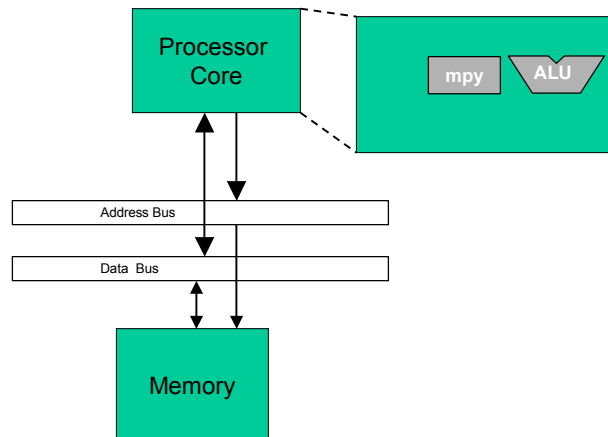
Execute row by row

10

ISLPED 2000, Tutorial. Ingrid Verbauwhede, Dave Garrett

Von Neumann machine

- One memory space



11

ISLPED 2000, Tutorial. Ingrid Verbauwhede, Dave Garrett

FIR on Von Neumann

Assume Von Neumann has multiply and accumulate instruction
(not necessarily the case)

Assume also that pipelining allows to execute the multiply and accumulate
in parallel with the read or write operations.

Then one tap needs 4 cycles:

1. read multiply-accumulate instruction
2. read data value from memory
3. read coefficient from memory
4. write data value to the next location in the delay line
(because for the next sample, all values are shifted by one location)

Memory bandwidth is crucial !!!

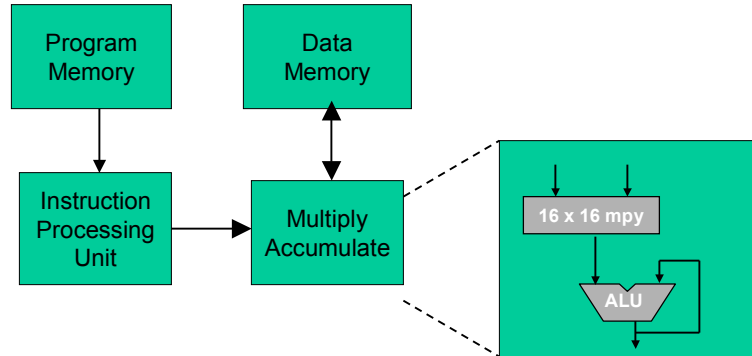
50 taps filter = 200 cycles!

12

ISLPED 2000, Tutorial. Ingrid Verbauwhede, Dave Garrett

Basic Harvard Architecture

- Separate data memory from program memory!

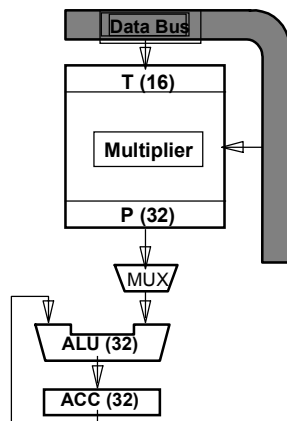


- Different from Von Neumann machine:
one address bus - one data bus - one memory space

13

ISLPED 2000, Tutorial. Ingrid Verbauwhede, Dave Garrett

TMS320C1x (1982)



Compute $Y = AX1 + BX2 + CX3 + DX4$

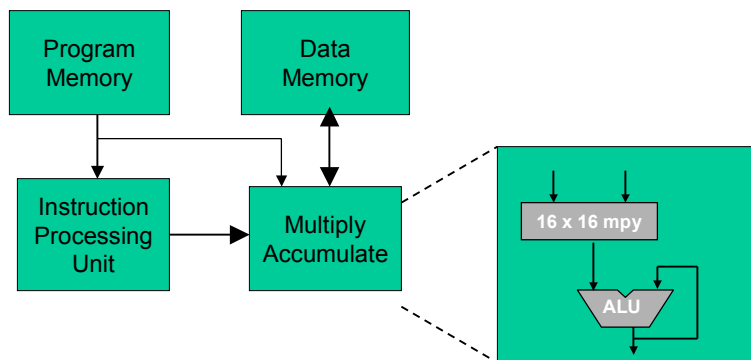
ZAC		ACC=0
LT	X1	T=X1
MPY	A	P=AX1
LTA	X2	ACC=AX1;T=X2
MPY	B	P=BX2
LTA	X3	ACC=AX1+BX2;T=X3
MPY	C	P=CX3
LTA	X4	ACC=AX1+BX2+CX3;T=X4
MPY	D	P=DX4
APAC		ACC=AX1+BX2+CX3+DX4
SACH	Y1	STORE 32-BIT RESULT
SACH	Y2	AT LOCATIONS Y1, Y2

- 50 taps = 104 cycles
- = Program ROM of 104 instructions

14

ISLPED 2000, Tutorial. Ingrid Verbauwhede, Dave Garrett

Modified Harvard Architecture

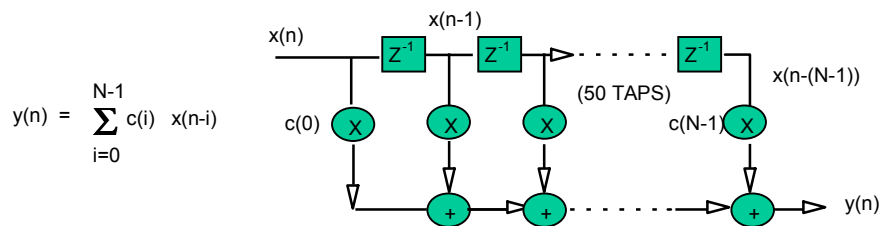


Program bus to get instruction
Or to get coefficients (often stored in ROM)

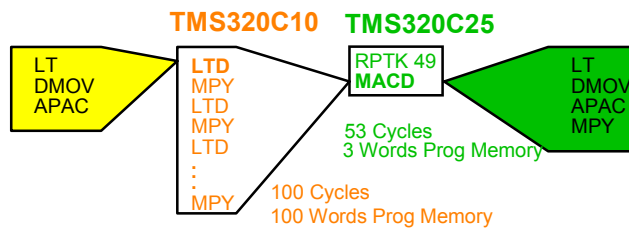
15

ISLPED 2000, Tutorial. Ingrid Verbauwhede, Dave Garrett

Same FIR: 53 cycles, 3 program words



Single Cycle Multiply - Accumulate!

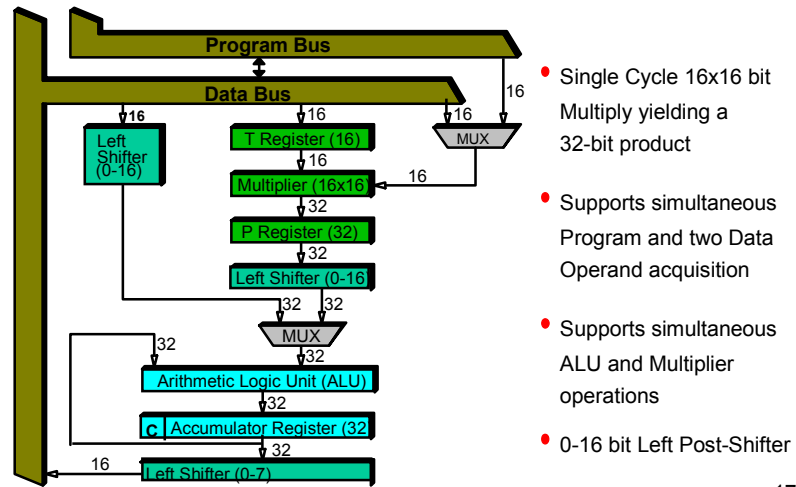


16

ISLPED 2000, Tutorial. Ingrid Verbauwhede, Dave Garrett

TMS320C2x (1986)

TMS320C2x Multiplier/ALU



ISLPED 2000, Tutorial. Ingrid Verbauwhede, Dave Garrett

17

FIR speed-up

- One output = 2N reads, N MAC's, 1 write
 - Two outputs = 4N reads, 2N MAC's, 2 writes
- Dual Mac Architecture with ONLY 2 data busses??*
- Read two 32-bit numbers instead of four 16-bit numbers
Solution by Lucent 16000 core with dual MAC
 - Run MAC at double frequency, read two 32-bit numbers
Solution by Matsushita
 - Insert delay register
Solution by Atmel's LODE (1996)
 - Use 3 busses
Solution by TI C55x (2000)

ISLPED 2000, Tutorial. Ingrid Verbauwhede, Dave Garrett

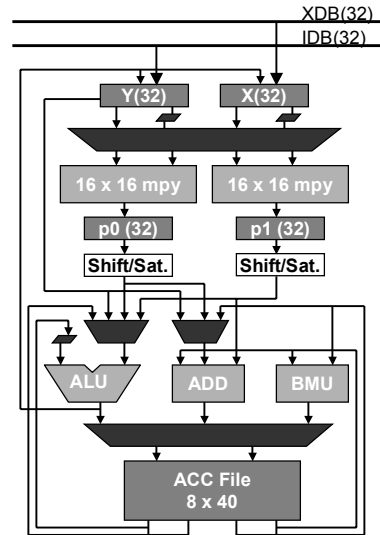
18

FIR on Lucent DSP16210

Inner loop of 32-tap FIR Filter

```
do 14 { //one instruction !
  a0=a0+p0+p1
  p0=xh*yh p1=xl*yl
  y=*r0++ x=*pt0++
}
```

- Outer Loop: 19 cycles, 38 bytes
1 cycle in inner loop
- 5 exec units used in inner loop
2 MACs per cycle
- Horizontal parallelism, one sample at a time
- 2G mobile wireless base-stations



Courtesy: Gareth Hughes, Bell Labs Australia

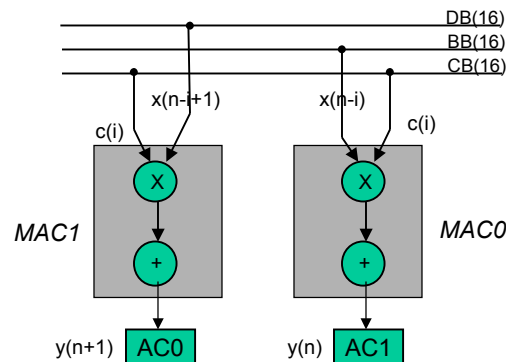
ISLPED 2000, Tutorial. Ingrid Verbauwhe, Dave Garrett

19

FIR on TI C55x (2000)

- FIR filter: two outputs in parallel with 3 busses

$$\begin{aligned}
 y(0) &= c(0)x(0) + c(1)x(-1) + c(2)x(-2) + \dots + c(N-1)x(1-N); \\
 y(1) &= c(0)x(1) + c(1)x(0) + c(2)x(-1) + \dots + c(N-1)x(2-N); \\
 y(2) &= c(0)x(2) + c(1)x(1) + c(2)x(0) + \dots + c(N-1)x(3-N); \\
 y(3) &= c(0)x(3) + c(1)x(2) + c(2)x(1) + \dots + c(N-1)x(4-N);
 \end{aligned}$$



ISLPED 2000, Tutorial. Ingrid Verbauwhe, Dave Garrett

20

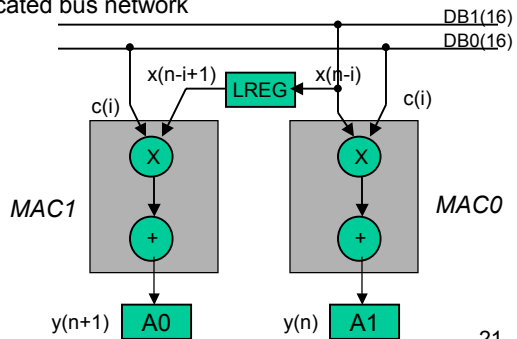
FIR on Lode (1996)

- FIR filter: two outputs in parallel with delay register

$$\begin{aligned}
 y(0) &= c(0)x(0) + c(1)x(-1) + c(2)x(-2) + \dots + c(N-1)x(1-N); \\
 y(1) &= c(0)x(1) + c(1)x(0) + c(2)x(-1) + \dots + c(N-1)x(2-N); \\
 y(2) &= c(0)x(2) + c(1)x(1) + c(2)x(0) + \dots + c(N-1)x(3-N); \\
 y(3) &= c(0)x(3) + c(1)x(2) + c(2)x(1) + \dots + c(N-1)x(4-N);
 \end{aligned}$$

- Two MAC units with dedicated bus network

- DB0 fetches coefficient
- DB1 fetches data
- LREG delays input data
- A0 stores $y(n)$ output
- A1 stores $y(n+1)$ output



21

ISLPED 2000, Tutorial. Ingrid Verbauwhede, Dave Garrett

Energy comparison

- Total energy for one output sample:

Energy	Single MAC	Dual MAC	Dual MAC 3 busses	Dual MAC with REG
No. of MAC operations	N	N	N	N
No of Memory reads	2N	2N	1.5N	N
No of Instruction Cycles	N	N/2	N/2	N/2

Adaptation of the datapath: MAC, DMAC

Adaptation of the memory architecture and bus network

Adaptation of the instruction set

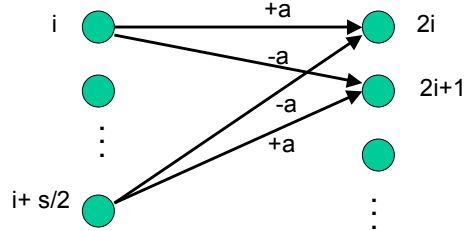
22

ISLPED 2000, Tutorial. Ingrid Verbauwhede, Dave Garrett

Compute Intensive function 2: Viterbi

- Viterbi butterfly

i = state index
 s = # of states = 2^{k-1}
 w = decoding window



- Basic equations:

$$d(2i) = \min \{ d(i) + a, d(i + s/2) - a \}$$

$$d(2i + 1) = \min \{ d(i) - a, d(i + s/2) + a \}$$

- Key operation: Add-Compare-Select (ACS)

- IS-95: $k = 9$, 256 states, $w = 192$, means $2^8 \times 192$ x (cycles for one ACS)

- Basic algorithm in Viterbi channel decoders and MLSE based receivers, modified version in turbo decoders.

23

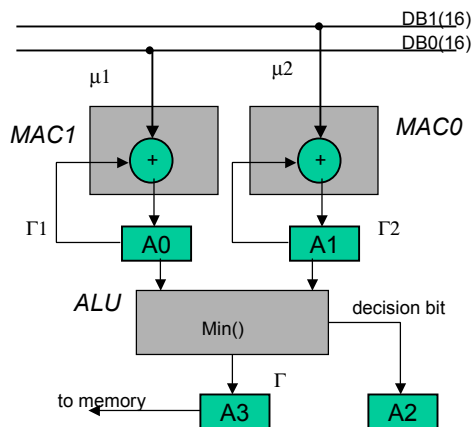
ISLPED 2000, Tutorial. Ingrid Verbauehede, Dave Garrett

Viterbi on Lode

- Two MAC units & ALU: Add-Compare-Select

$$\Gamma = \min [(\Gamma_1 + \mu_1), (\Gamma_2 + \mu_2)]$$

- DMAC operates as dual add/subtract unit
- ALU finds minimum
- Shortest distance saved
- Path indicator saved
- 4 cycles / butterfly



24

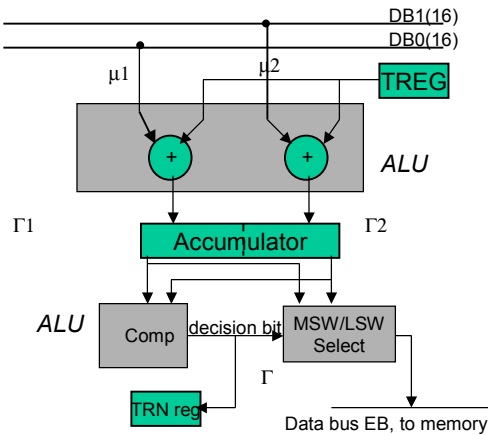
ISLPED 2000, Tutorial. Ingrid Verbauehede, Dave Garrett

Viterbi on TIC54x

- ALU and CSSU: CMPS instruction

$$\Gamma = \min [(\Gamma_1 + \mu_1), (\Gamma_2 + \mu_2)]$$

- ALU splits in 16 bit halves
- ACC splits in half
- Shortest distance saved
- CSSU compares halves
- Path indicator saved
- 4 cycles / butterfly



Source: TI Application Report, Viterbi Decoding in the TMS320C54x family, document SPRA071

ISLPED 2000, Tutorial. Ingrid Verbauwhede, Dave Garrett

25

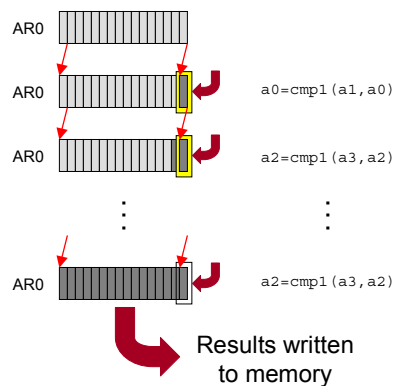
Viterbi on LU DSP16210

GSM (K=5, 16 states)

```
do 8 {
  a0=a4+y a1=a5-y *r3++=a0h
  a2=a4-y a3=a5+y *r5++=a2h
  a0=cmp1(a1,a0) yh=*r0 r0=r1+j j=k k=*pt1++
  a2=cmp1(a3,a2) a4_5h=*pt0++
}
```

- Hardware support for Viterbi algorithm:
 - ACS calculations are efficient
 - Minimal overhead
- 4 cycles per butterfly
 - 32 cycles per GSM timeslot.

- Comparison functions store ACS decision bits:



Courtesy: Gareth Hughes, Bell Labs Australia

ISLPED 2000, Tutorial. Ingrid Verbauwhede, Dave Garrett

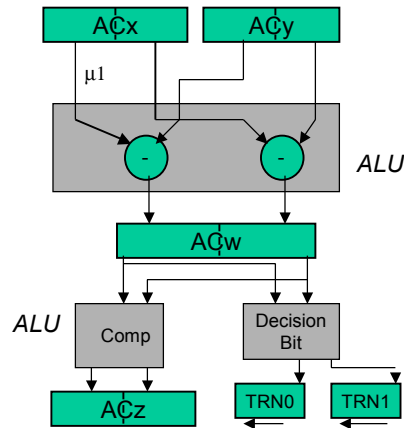
26

Turbo coding on TIC55x

- ALU and CSSU: *max_diff* instruction

Maxdiff ACx, ACy, ACz, ACw

- ALU splits in 16 bit halves
- 4 ACC's splits in half
- differences stored in ACw halves
- Max stored in ACz halves
- Path indicators saved in TRN's



Source: TMS320C55x DSP Mnemonic Instruction Set Ref. Guide, document SPRU374C

27

ISLPED 2000, Tutorial. Ingrid Verbauwhede, Dave Garrett

Case study: LPC Speech Coder

Main computation modules:

- FIR's, autocorrelation, Levinson-Durbin algo

Platform	Clock Freq.	Cycles (MIPS)	Instruction width	Energy/sample	Technology	Power Supply
TI C5402	12 MHz	240K	16	267 nJ	0.18 μm^b	1.8 V core 3.3 V I/O
TI C5510	5 MHz	120K	8 - 48	20.1 nJ	< 0.18 μm^b	1.6 V core 3.3 V I/O
TI C6211	150 MHz ^a	30K	256	1800 nJ	0.18 μm	1.8 V core 3.3 V I/O
OCAPI	600 KHz	11K		13.4 nJ	0.25 μm	2.5 V
AR T	150KHz	3K	120	27 nJ	0.35 μm	3.3 V

a: accommodates around 75 channels
b: estimates

28

ISLPED 2000, Tutorial. Ingrid Verbauwhede, Dave Garrett

Low Power DSP's

DSP 1600 Core

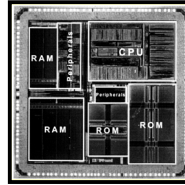
(Lucent - 1609 low cost consumer 16-bit)



- 0.35 μ 3LM CMOS
- 80 M 16b MAC/s at 3.3V
- 1.4 mW/MHz at 3.3V
- 30 μ W stand-by power

C54x 1V DSP

(Texas Instruments - ISSCC 1997)



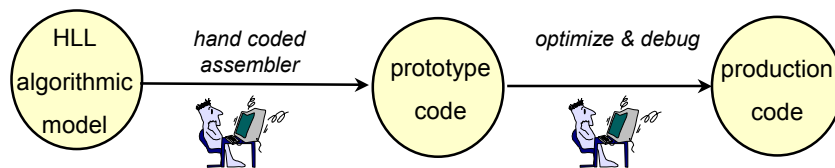
- 0.25 μ 3LM CMOS
- 63 M 16b MAC/s at 1.0V
- 0.21 mW/MHz at 1.0V
- 4.0 mW stand-by power
- Dual V_t process

29

ISLPED 2000, Tutorial. Ingrid Verbauwhede, Dave Garrett

BUT: DSP Software Development

- Complex DSP architecture not amenable to compiler technology
- Algorithms are modeled in high level language (e.g. C++)
- Solutions are implemented and debugged in hand-optimized assembler - large development effort with minimal tool support



- ➔ Long, frustrating time to market
- ➔ Fragile legacy code

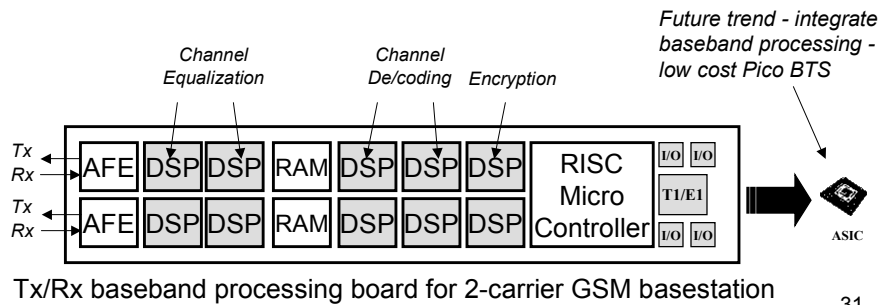
Widely used in handhelds, but change in basestations ➔ **Part II**

30

ISLPED 2000, Tutorial. Ingrid Verbauwhede, Dave Garrett

2G Basestation Baseband Processing

- Multiple DSPs used for baseband processing.
- RISC Microcontroller for timing, framing, I/O control
- Software upgradable over the network
- DSPs dominate cost and power consumption



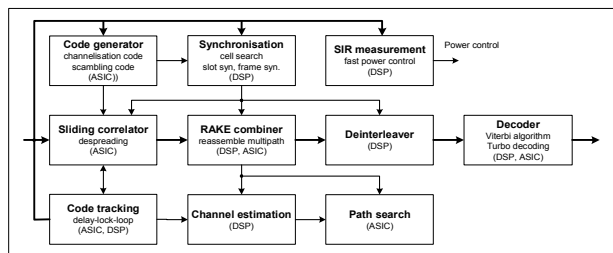
ISLPED 2000, Tutorial. Ingrid Verbauwhe, Dave Garrett

3G Basestation Baseband Processing

Increased DSP performance needed in next-generation basestation

- Increased Receiver Algorithm Sensitivity
- Antenna Arrays - Smart Antennas
- Multi-Standard Basestations using Software Radio Architecture
- 3G - constraint length 9, rate 1/2 convolutional coding for voice.
- 3G - constraint length 4, Turbo codes for data

High Performance DSPs
+ Custom Logic needed
for 3G (Viterbi decoding
and Turbo decoding)

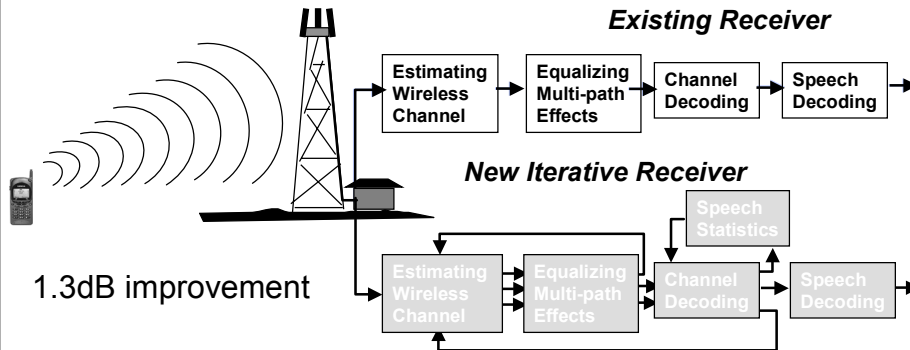


Courtesy: Bing Xu: Bell Labs Australia

ISLPED 2000, Tutorial. Ingrid Verbauwhe, Dave Garrett

Receiver Algorithms for GSM Basestation

- Enhanced Receiver Sensitivity
- Larger Cells in Suburban Areas = Reduced network cost
- Mobile transmits with less power = Increased battery life

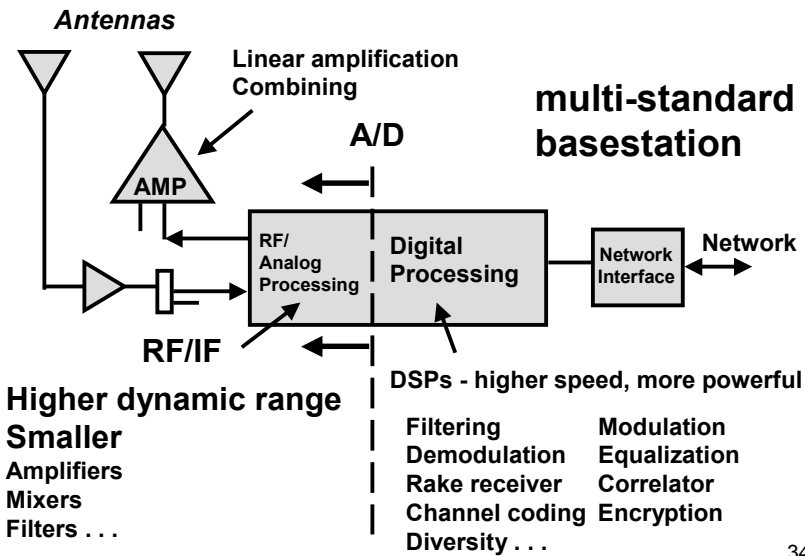


Challenge - requires 6x DSP MIPS of existing receiver in basestation

Courtesy: Magnus Sandell: Bell Labs UK 33

ISLPED 2000, Tutorial. Ingrid Verbauwhe, Dave Garrett

Digital Radio Trends - Software Radio



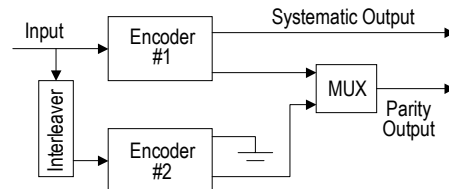
34

ISLPED 2000, Tutorial. Ingrid Verbauwhe, Dave Garrett

Turbo Codes

For 3G Wireless (UMTS and CDMA2000)

- Voice service: BER requirement 10^{-3}
- Data service: BER requirement 10^{-5}
- *Parallel* concatenation of convolutional codes is used to give the codes structure so they can be decoded
- Pseudorandom interleaving is used to give the codes performance which approaches that for random coding
- Resulting encoder structure: Two Recursive Systematic Convolutional(RSC) Codes

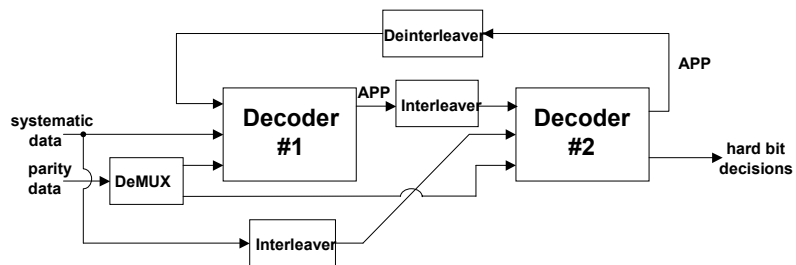


35

ISLPED 2000, Tutorial. Ingrid Verbauwhede, Dave Garrett

Turbo Decoding

- Key idea: iterative decoding (up to 10 iterations for 3G)
- There is one decoder for each elementary encoder.
- Each decoder estimates the *a-posteriori probability* (APP) of each data bit.
- The APP's are used as *a priori* information by the other decoder.



36

ISLPED 2000, Tutorial. Ingrid Verbauwhede, Dave Garrett

Soft-Output Decoding Algorithms

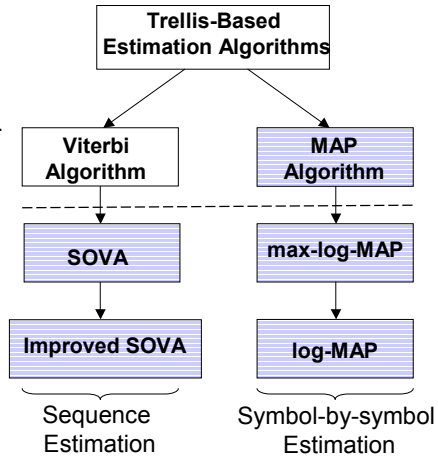
Requirements for Turbo:

- Accept Soft-Inputs in the form of a priori probabilities (APP)
- Produce APP estimates of the data.
- "Soft-Input Soft-Output"

Today's High-performance DSPs are highly MAC-focussed (for filtering in modem applications). Some DSPs provide hardware support for efficient implementation of Viterbi - few support SOVA or log-MAP

Iterative channel estimation also uses Soft-Input Soft-Output decoders.

SOVA and log-MAP use modified Add-Compare-Select operations - not only select the maximum path metric - but also need to keep the difference.

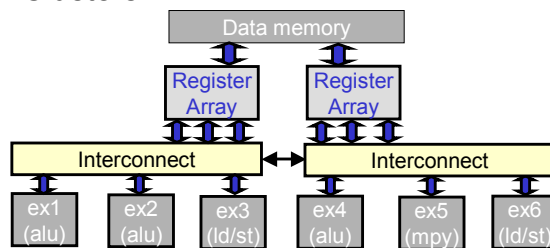


37

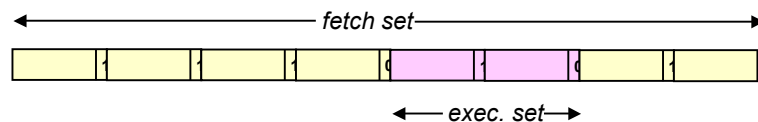
ISLPED 2000, Tutorial. Ingrid Verbauwhede, Dave Garrett

Explicitly Parallel Instruction Computing

Execution Clusters



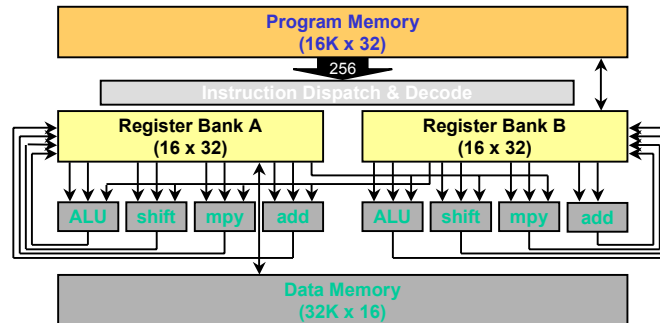
Execution Sets



38

ISLPED 2000, Tutorial. Ingrid Verbauwhede, Dave Garrett

Texas Instruments 'C6201



- 8-way VLIW with two execution clusters
- 256 bit (8x32) instruction fetch with variable length execute set
- Each 32 bit instruction individually predicated
- 11 stage pipeline
- 1600 MIPS, 400 MMACs @ 200 MHz

39

ISLPED 2000, Tutorial. Ingrid Verbauwhede, Dave Garrett

FIR Filter on TI 'C6x

Hand-coded assembly: 32-tap FIR filter

```

loop:
    ldw      .d1t1    *a4++,a5
    ||      ldw      .d2t2    *b4++,b5
    || [b0] sub      .s2     b0,1,b0
    || [b0] b        .s1     loop
    ||      mpy      .m1x    a5,b5,a6
    ||      mpyh     .m2x    a5,b5,b6
    ||      add      .11     a7,a6,a7
    ||      add      .12     b7,b6,b7
    
```

- Outer Loop: 23 cycles, 180 bytes
 - 1 cycle in inner loop
- All 8 exec units used in inner loop - maximum efficiency
 - 2 MACs per cycle

Assembly syntax more difficult to learn.

Hard to get full use of all 8 execution units at once.

Software pipelining difficult to implement, and requires longer prolog/epilog (larger code size).

Courtesy: Gareth Hughes: Bell Labs Australia

40

ISLPED 2000, Tutorial. Ingrid Verbauwhede, Dave Garrett

Viterbi on TI 'C6x

3-cycle 2-ACS Inner-Loop

```

LOOP:
[bl]      b      .s1  LOOP
[bl]      sub    .s2  b1,1,b1
[!a2]     sth    .d1  b12,++a6[8]
[!a2]     add    .d2  b0,b14,b14
[!a2]     cmpgt  .l1  a11,a10,a1
[!a2]     cmpgt  .l2  b11,b10,b0
[!a2]     mpy    .mlx 1,b5,a4

[a2]      sub    .s1  a2,1,a2
[!a2]     sth    .d1  a12,*a6++
[!a1]     add    .s2  2,b0,b0
[!b0]     mpy    .m2  1,b11,b12
[!b0]     mpy    .m1  1,a10,a12
[!b0]     sub    .l2x a7,b5,b10
[!b0]     ldh    .d2  ++b9,b5

[a1]      shl    .s2  b14,2,b14
[!a1]     mpy    .m1  1,a11,a12
[!a1]     add    .s1  a7,a4,a10
[!a1]     sub    .l1x b13,a4,a11
[!a1]     add    .l2  b13,b5,b11
[!a1]     mpy    .m2  1,b10,b12
[!a1]     ldh    .d2  *b4++[2],a7
[!a1]     ldh    .d1  *a5++[2],b13
; end of LOOP
    
```

- 16-state Viterbi decoder for GSM
 - 3 cycles per butterfly
 - 32 cycles per GSM timeslot (8 butterflies)
 - MPY instructions used to move data

x 8

Cycle	0	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30
C1	STHnew0	LDHnew0	STHnew0	STHnew0	LDHnew0	STHnew0	STHnew0	LDHnew0	STHnew0	STHnew0	LDHnew0	STHnew0	STHnew0	LDHnew0	STHnew0	STHnew0
C2	ADD	LDHnew1	LDHnew1	ADD	LDHnew1	LDHnew1	ADD	LDHnew1	LDHnew1	ADD	LDHnew1	LDHnew1	ADD	LDHnew1	LDHnew1	ADD
M1	MPYnew0	MPYnew0	MPYnew0	MPYnew0	MPYnew0	MPYnew0	MPYnew0	MPYnew0	MPYnew0	MPYnew0	MPYnew0	MPYnew0	MPYnew0	MPYnew0	MPYnew0	MPYnew0
M2	MPYnew1	MPYnew1	MPYnew1	MPYnew1	MPYnew1	MPYnew1	MPYnew1	MPYnew1	MPYnew1	MPYnew1	MPYnew1	MPYnew1	MPYnew1	MPYnew1	MPYnew1	MPYnew1
L1	CMPTnew0	SUBnew0	CMPTnew0	SUBnew0	CMPTnew0	SUBnew0	CMPTnew0	SUBnew0	CMPTnew0	SUBnew0	CMPTnew0	SUBnew0	CMPTnew0	SUBnew0	CMPTnew0	SUBnew0
L2	CMPTnew1	ADDnew1	SUBnew1	CMPTnew1	ADDnew1	SUBnew1	CMPTnew1	ADDnew1	SUBnew1	CMPTnew1	ADDnew1	SUBnew1	CMPTnew1	ADDnew1	SUBnew1	CMPTnew1
S1	B.LOOP	ADDnew2	SUBnew2	B.LOOP	ADDnew2	SUBnew2	B.LOOP	ADDnew2	SUBnew2	B.LOOP	ADDnew2	SUBnew2	B.LOOP	ADDnew2	SUBnew2	B.LOOP
S2	SUB	SHL	*ADDnew3	SUB	SHL	*ADDnew3	SUB	SHL	*ADDnew3	SUB	SHL	*ADDnew3	SUB	SHL	*ADDnew3	SUB

Cycle	1	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31
C1	STHnew1	STHnew1	LDHnew1	STHnew1	STHnew1	LDHnew1	STHnew1	STHnew1	LDHnew1	STHnew1	STHnew1	LDHnew1	STHnew1	STHnew1	LDHnew1	STHnew1
C2	LDHnew0	ADD	LDHnew0	LDHnew0	ADD	LDHnew0	LDHnew0	ADD	LDHnew0	LDHnew0	ADD	LDHnew0	LDHnew0	ADD	LDHnew0	LDHnew0
M1	MPYnew1	MPYnew1	MPYnew1	MPYnew1	MPYnew1	MPYnew1	MPYnew1	MPYnew1	MPYnew1	MPYnew1	MPYnew1	MPYnew1	MPYnew1	MPYnew1	MPYnew1	MPYnew1
M2	MPYnew2	MPYnew2	MPYnew2	MPYnew2	MPYnew2	MPYnew2	MPYnew2	MPYnew2	MPYnew2	MPYnew2	MPYnew2	MPYnew2	MPYnew2	MPYnew2	MPYnew2	MPYnew2
L1	MPYnew3	CMPTnew1	SUBnew1	CMPTnew1	SUBnew1	CMPTnew1	SUBnew1	CMPTnew1	SUBnew1	CMPTnew1	SUBnew1	CMPTnew1	SUBnew1	CMPTnew1	SUBnew1	CMPTnew1
L2	SUBnew0	CMPTnew2	ADDnew2	SUBnew0	CMPTnew2	ADDnew2	SUBnew0	CMPTnew2	ADDnew2	SUBnew0	CMPTnew2	ADDnew2	SUBnew0	CMPTnew2	ADDnew2	SUBnew0
S1	SUB	B.LOOP	ADDnew3	SUB	B.LOOP	ADDnew3	SUB	B.LOOP	ADDnew3	SUB	B.LOOP	ADDnew3	SUB	B.LOOP	ADDnew3	SUB
S2	*ADDnew4	SUB	SHL	*ADDnew4	SUB	SHL	*ADDnew4	SUB	SHL	*ADDnew4	SUB	SHL	*ADDnew4	SUB	SHL	*ADDnew4

Utilization of execution units in Viterbi decoder

ISLPED 2000, Tutorial. Ingrid Verbauehede, Dave Garrett

New >1GHz C64x DSP Architecture

- Similar Architecture as C62x
- Additional 32 registers
- Additional 14 execute units and load/store units

Clock Freq. (MHz) 600 - 1100
 16-b MMACs 2400 - 4400

No power numbers are available at present

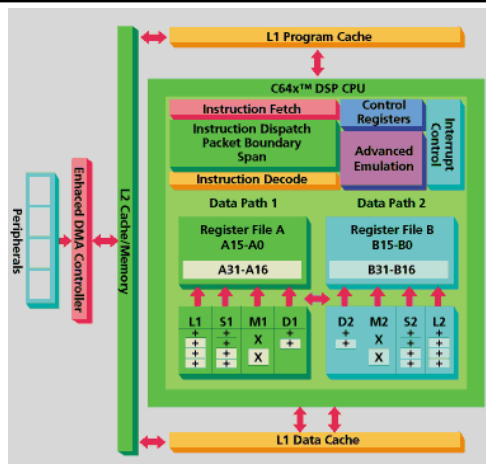
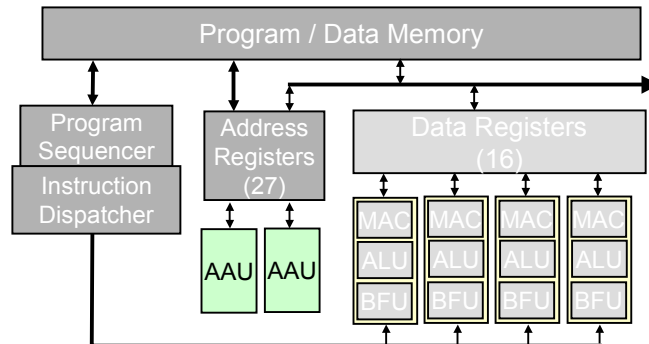


Diagram from www.ti.com/sc/docs/products/dsp/newcores/c64x.htm

ISLPED 2000, Tutorial. Ingrid Verbauehede, Dave Garrett

Lucent / Motorola Star*Core SC140

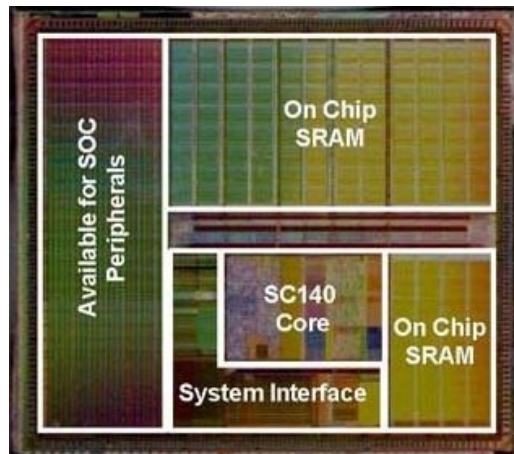


- 6-way VLIW with 128 bit (8x16) instruction fetch
- Prefix instructions for high performance without sacrificing code density
- Each execution set (parallel instructions + prefix) predicated
- 5 stage pipeline
- 1800 MIPS, 1200 MMACs @ 300 MHz

43

ISLPED 2000, Tutorial. Ingrid Verbauwhede, Dave Garrett

Starcore Die Photo



Low Power DSP Chip for
2.5G and 3G Mobile
Wireless Handsets

Low Power is achieved by:

- Static CMOS logic
- DSP Instruction Set (more DSP-like than RISC-like, e.g. MAC rather than MULT, ADD)

0.11mA/MMAC (198mW total) @ 1.5V
0.066mA/MMAC (28mW total) @ 0.9V

44

ISLPED 2000, Tutorial. Ingrid Verbauwhede, Dave Garrett

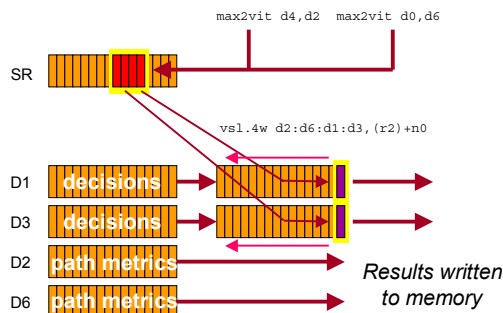
Viterbi on Star*Core

GSM (K=5, 16 states)

```
[ move.2l (r0)+,d0:d1  move.2l (r1)+,d1:d2 ]
[ add2 d0,d4          sub2 d6,d2          ]
[ sub2 d4,d0          add2 d2,d6          ]
[ max2vit d4,d2      max2vit d0,d6      ] x 4
[ vs1.4w d2:d6:d1:d3,(r2)+n0
  vs1.4f d2:d6:d1:d3,(r3)+n0          ]
```

- Decision bits are manually stored using the Viterbi Shift Left (VSL) instruction:

- Hardware support for Viterbi algorithm:
 - `max2vit` instruction.
 - `vs1` instruction
- 1 cycle per butterfly through software-pipelining

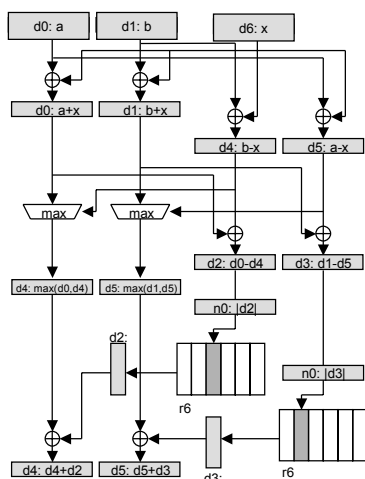


Courtesy: Gareth Hughes: Bell Labs Australia

ISLPED 2000, Tutorial. Ingrid Verbauwede, Dave Garrett

45

Log-MAP on Star*Core



Star*Core code for log-MAP Butterfly

```
Cycle 1 move.w (r0)+,d0  move.w (r1)+,d1
Cycle 2 add d0,d6,d0    sub d6,d0,d5
Cycle 3 sub d6,d1,d4   add d1,d6,d1
Cycle 4 sub d0,d4,d2   sub d1,d5,d3
Cycle 5 max d0,d4     max d1,d5
Cycle 6 abs d2       abs d3
Cycle 7 move.l d2,n0
Cycle 8 move.l d3,n0  move.w (r6+n0),d2
Cycle 9 add d4,d2,d4  move.w (r6+n0),d3
Cycle 10 add d5,d3,d5
Cycle 11 move.2w d4:d5,(r2)+
```

This code uses 2 of the 4 ALUs and can be software pipelined to achieve 6 cycles per LOG-MAP Butterfly

Courtesy: Gareth Hughes: Bell Labs Australia

ISLPED 2000, Tutorial. Ingrid Verbauwede, Dave Garrett

46

Parallel DSP Architectures

Arch.	Parallelism	Compile?	Power ?
S/scalar	Dynamic instruction level		xx
VLIW	Static instruction level		x
SIMD	Highly regular, data dependent	xx	
MIMD	Task level	x	

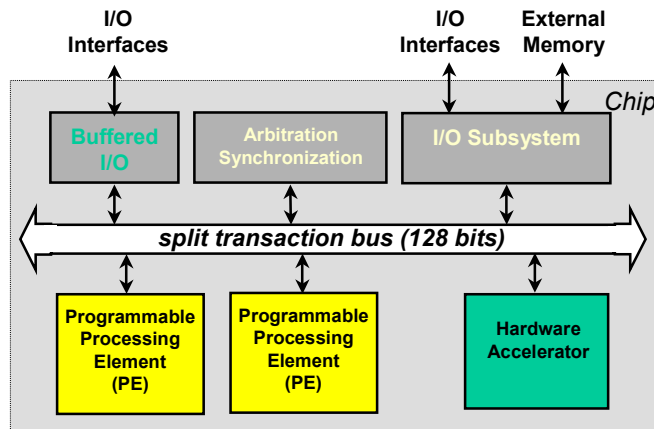
- MIMD with VLIW / SIMD provides high order parallel execution

The future of high performance DSPs is MIMD

47

ISLPED 2000, Tutorial. Ingrid Verbauwhede, Dave Garrett

Daytona: A Multiprocessor DSP Architecture



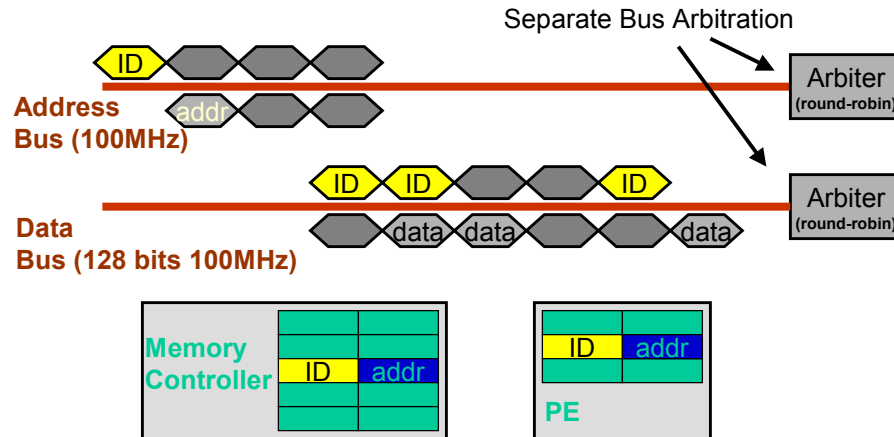
- Scalable Architecture - multiple programmable DSPs on a single chip
- 1 Bus supports different programmable DSPs and Microcontrollers

48

ISLPED 2000, Tutorial. Ingrid Verbauwhede, Dave Garrett

Split Transaction Bus

- Separate Address and Data busses - each with pipelined protocol
- Multiple outstanding transactions - varying size/priority



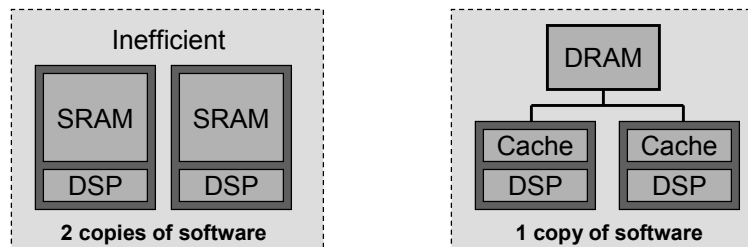
49

ISLPED 2000, Tutorial. Ingrid Verbauwhede, Dave Garrett

Memory Hierarchy in MIMD DSPs

- Multiple copies of 1 application (e.g. odd/even slot channel equalisation)
 - Multiple copies of same software - Shared memory multiprocessing

Flat Memory Architecture vs. Hierarchical Memory Architecture

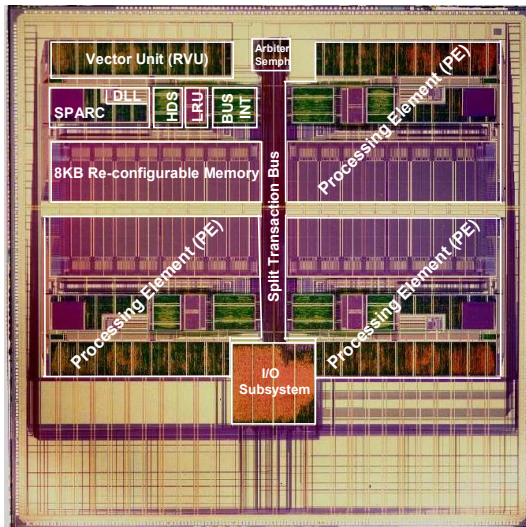


- Mix of different applications (e.g. equalisation, convolutional decoding)
 - Heterogenous mix of applications

50

ISLPED 2000, Tutorial. Ingrid Verbauwhede, Dave Garrett

Photomicrograph of Daytona Test Chip



Test Chip from Bell Labs Research.

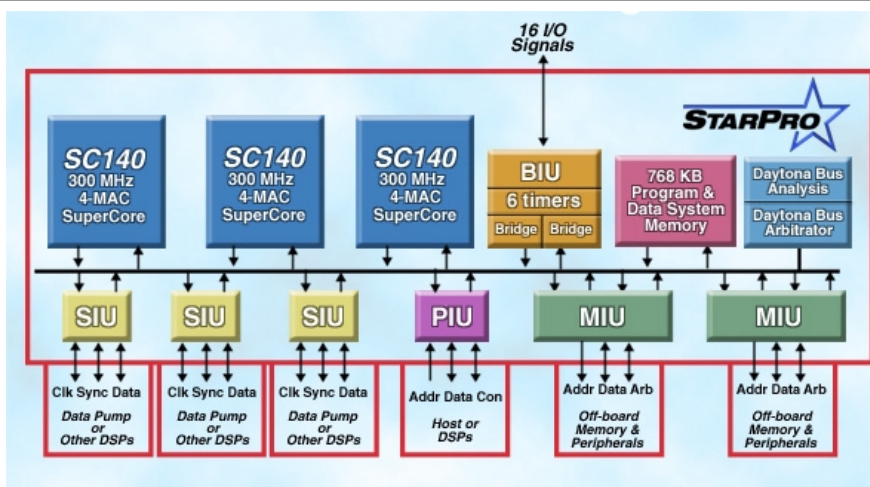
MIMD DSP Daytona platform has been adopted by Lucent Microelectronics as SOC platform

Paper 4.2, ISSCC2000

51

ISLPED 2000, Tutorial. Ingrid Verbauwhede, Dave Garrett

StarPro - Daytona-based MIMD Starcore DSP



3600 MMACs @ 300MHz, 3G Basestation Chip.

52

ISLPED 2000, Tutorial. Ingrid Verbauwhede, Dave Garrett

Conclusions

- Power savings at the **architectural** level can be huge
- Tune all components of the architecture to the application:
 - datapaths
 - instruction set
 - **memory architecture and bandwidth**
- Complexity of 2nd, 3rd generation cellular standards:
 - require multiple processors
 - compiler friendly
- **How to combine compiler friendliness with power efficiency?**

Web page: www.ee.ucla.edu/~ingrid/Presentations/presentations.html

53

ISLPED 2000, Tutorial. Ingrid Verbauwhede, Dave Garrett

References

- [1] P. Lapsley, J. Bier, A. Shoham, E. Lee, "DSP Processor Fundamentals," IEEE Press, New York, 1997.
- [2] D. Skillikorn, "A Taxonomy for Computer Architectures," *Computer Magazine*, Nov. 1988.
- [3] H. Kabuo, M. Okamoto, I. Tanaka, H. Yasoshima, S. Marui, M. Yamasaki, T. Sugimura, K. Ueda, T. Ishikawa, H. Suzuki, R. Asahi, "An 80 MOPS-Peak High-Speed and Low-Power-Consumption 16-b Digital Signal Processor," *IEEE Journal of Solid-State Circuits*, Vol. 31, No. 4, April 1996, pg. 494-503.
- [4] E. A. Lee, D. G. Messerschmitt, *Digital communication*, Boston: Kluwer Academic Publishers, 1988.
- [5] W. Lee et al., "A 1V DSP for Wireless Communications," *Proceedings IEEE International Solid-State Circuits Conference*, pp. 92-93, February 1997.
- [6] S. Lin, and J. Costello Jr., *Error Control Coding: Fundamentals and applications*, Prentice Hall, New Jersey, 1983
- [7] Lucent 16000, <http://www.lucent.com/micro/> or <http://www.lucent.dk/micro/dsp16000/>
- [8] Thomas Parsons, *Voice and Speech Processing*, McGraw-Hill Book Company, New York, 1987.
- [9] TMS320C54x User's Guide, available from the Texas Instruments Literature Response Center.
- [10] I. Verbauwhede, M. Touriguan, "A Low Power DSP Engine for Wireless Communications," *Journal of VLSI Signal Processing* 18, pg. 177-186, 1998, Kluwer Academic Publishers.
- [11] I. Verbauwhede, M. Touriguan, "Wireless digital signal processors," Chapter in *Digital Signal Processing for Multimedia Systems*, Edited by K.K. Parhi, T. Nishitani, Publisher: Marcel Dekker, New York, 1999.
- [12] M. Okamoto, K. Stone, T. Sawai, H. Kabuo, S. Marui, M. Yamasaki, Y. Uto, Y. Sugisawa, Y. Sasagawa, T. Ishikawa, H. Suzuki, N. Minamida, R. Yamanaka, K. Ueda, "A High Performance DSP Architecture for Next Generation Mobile Phone Systems," 1998 *IEEE DSP Workshop*.
- [13] Lode specifications, available from www.atmel.com
- [14] M.W. Oliphant, "The Mobile Phone meets the Internet", *IEEE Spectrum* pp. 20-28, Aug. 1999.
- [15] L. C. Godara, "Application of Antenna Arrays to Mobile Communications: Part 1", *Proc. IEEE*, Vol 85, No. 7. pp 1031-1060, July 97

54

ISLPED 2000, Tutorial. Ingrid Verbauwhede, Dave Garrett

References (cont)

- [16] G. D. Forney, Jr., "Maximum Likelihood Sequence Estimation of Digital Sequences in the Presence of Intersymbol Interference", *IEEE Trans. Inform. Theory*, V IT-18, pp. 363-378, May 1972.
- [17] C. Berrou, A. Glavieux, P. Thitimajshima, "Near Shannon Limit Error-Correcting Coding and Decoding: Turbo-Codes (1)", *Proc. ICC'93*, May 1993.
- [18] J. Hagenauer, P. Hoeher, "A Viterbi Algorithm with Soft-Decision Outputs and its Applications", *Proc. Globecom 89*, Nov. 1989, pp.47.1.1-47.1.7
- [19] L. Bahl, J. Cocke, F. Jelinek, J. Raviv, "Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate", *IEEE Trans. Inform. Theory*, V IT-20, pp. 284-287, Mar. 1974.
- [20] J. Turley, H. Hakkaraainen, "TI's new 'C6x DSP Screams at 1600 MIPS", *Microprocessor Report*, Vol 11, No. 2, pp 14, Feb 1997
- [21] "Starcore Launched First Architecture", *Microprocessor Report*, V12, No. 14, pp 22, Oct 1998
- [22] B. Ackland & P. D'Arcy, "A New Generation of DSP Architectures", *Proc. IEEE CICC99*, Paper 25.1.1
- [23] J. Williams, K.J. Singh, C.J. Nicol, B. Ackland, "A 3.2 GOPs Multiprocessor DSP for Communication Applications", *Proc. IEEE ISSCC2000*, Paper 4.2