Processor Versatility (Flexibility),
an attempt at definition and quantification

MPSoC 2013, Otsu, Japan, July 17

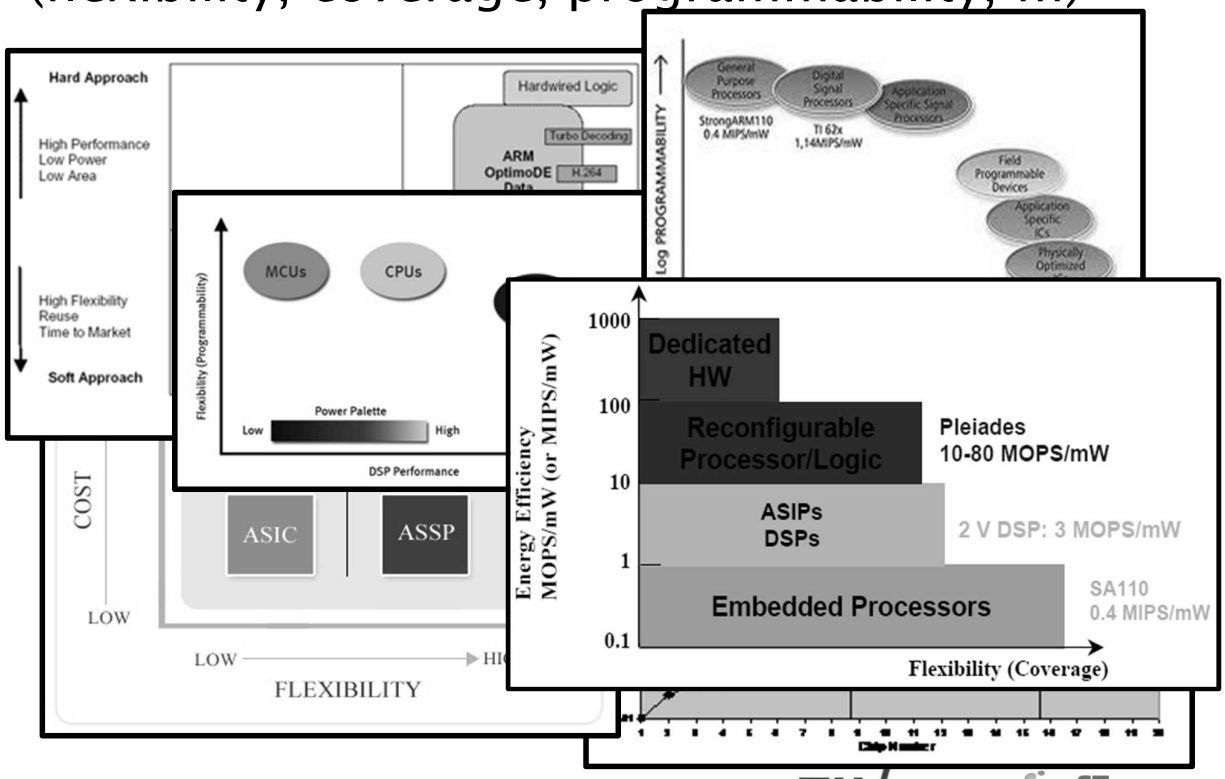
Kees van Berkel

ITEA 2
INFORMATION TECHNOLOGY FOR EUROPEAN ADVANCEMENT
/ MANY

professor @ TU/e

fellow @ ERICSSON
TAKING YOU FORWARD

Processor versatility
(flexibility, coverage, programmability, …)



Versatility

- the quality or state of being versatile,

where versatile means:

- changing or fluctuating readily : variable;
- embracing a variety of subjects, fields, or skills;
- turning with ease from one thing to another;
- having many uses or applications.

A processor is versatile (relative to another) if it:

- has many uses or applications;
- embraces a variety of algorithms, data types, etc
- is easy to program;
- can change readily from one task to another.

Processor versatility (1)

- How can we quantify processor versatility?
- How do various processors measure up in terms of versatility? (CPU, GPU, DSP, SIMD DSP, ASIP, …)
- What does versatility mean for costs and power?
- How can we quantify MPSoC versatility?

Processor versatility (2)

Choose a "unit of work" for which the processor hardware can be configured

- how much information is required to specify that unit of work?
- how much useful work is specified by that unit of work?

Proposal: $$versatility = \frac{amount\ infomation\ to\ specify\ a\ unit\ of\ work}{amount\ of\ useful\ work\ specified}$$

For processors we choose the instruction as a unit of work:

Proposal: $$versatility = \frac{average\ instruction\ size}{number\ of\ useful\ operations\ per\ instruction}$$

Note: versatility thus defined is a property of the *instruction set architecture* (ISA), and not a property of a specific ISA implementation.

Processor versatility (3)

… but a 64-bit machine is more versatile than a 32-bit machine …

Proposal: $$versatility = \frac{average\ instruction\ size \times \boldsymbol{precision}}{number\ of\ useful\ operations\ per\ instruction}$$

Where:

- average instruction size is in bits, e.g. 32 bit;
- precision is in words, typically ½, 1, or 2;
- number of useful operations per instruction is a real number >0.

Notes:

- extend ISA for same throughput ⇒ versatility ↑
- specify more work per same-sized instruction
  ⇒ less resolution among operations ⇒ versatility ↓

Processor versatility (4)

Processor versatility is a property of the *instruction-set architecture*, and not of a specific implementation of that ISA.

Hence, processor versatility and is not affected by:

- pipelining, branch prediction, caching, multi-issue, multi-core, etc.

That is, processor versatility is orthogonal to throughput $\frac{f_{clock}}{CPI}$

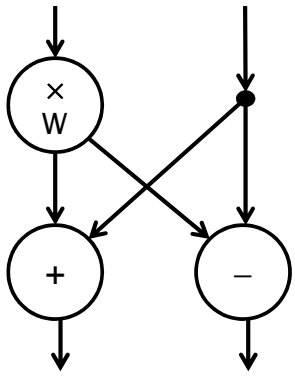
Processor versatility (5)

A complex FFT is used to measure the versatility of a range of processors

- FFT data is available for a large variety of different architectures;
- FFT code typically hand optimized assembly; reduces compiler impact.

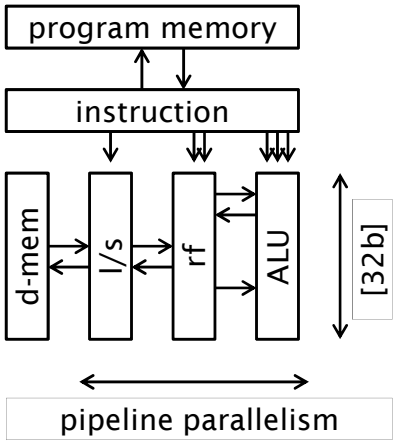
In order to measure the number of operations, we *assume* an FFT based on *radix-2 butterflies* (following http://www.fftw.org/speed/)

- 1 *complex* radix-2 butterfly = 8 operations (4 multiplications + 4 additions/subtractions)
- $ops\ per\ instruction = \frac{8\ N/2\ \log_2 N}{\#cycles\ per\ FFT} \times CPI$
- Where *N* is the FFT block size, e.g. 256 and *CPI* denotes Cycles Per Instruction



Note: available CPI data usually typical, i.e. not FFT specific.

A RISC processor: MIPS 4K

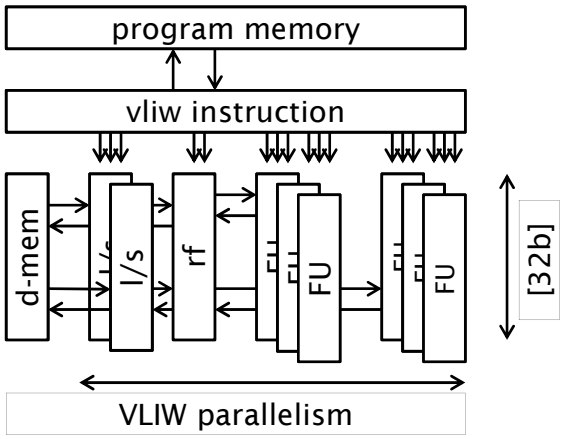


- max 1 results per cycle
- 1load/store unit, 1 ALU
- 32 g.p. registers, 32-bit*
- CPI: ≈2

ISA	MIPS 4k
precision	1
instruction width	32
ops/instruction	0.74
versatility	43

*MIPS III ISA is 64-bit; FFT data based on 32 bit

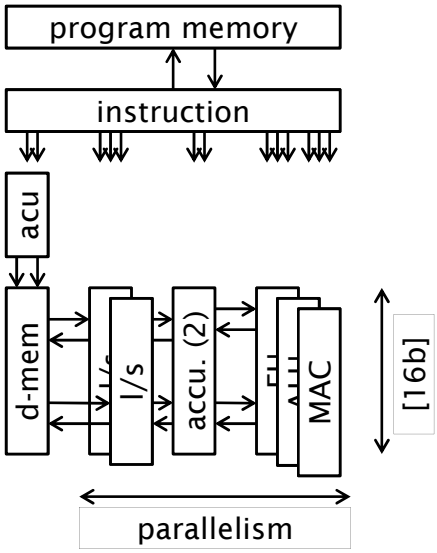
VLIW / TriMedia-1000



- VLIW: 5 issue slots
- up to 5 results per cycle
- 28 FUs, incl. 2 load/store
- 128 registers, 32-bit

ISA	Trimedia-1000
precision	1
instruction width	120
ops/instruction	3.8
versatility	32

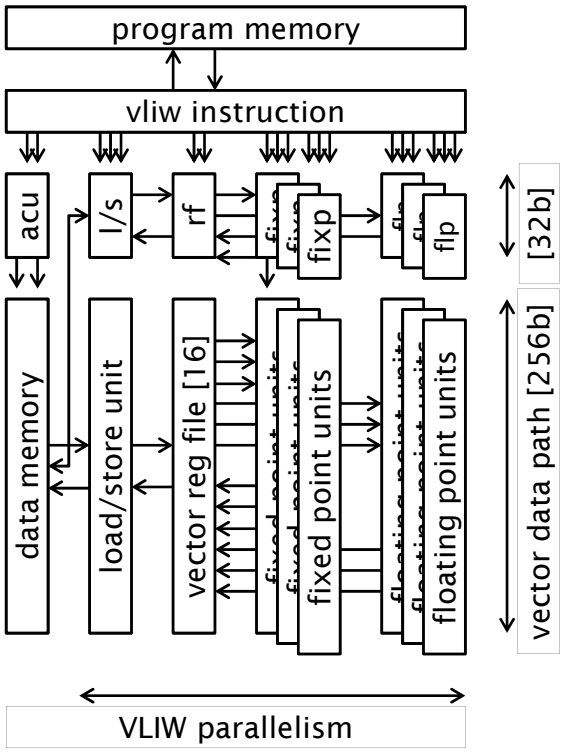
DSP / TMS320C54



- dual-access data memory
- + program memory bus
- single-cycle MAC + ALU
- 16-bit fixed point data
- 2 accumulators (40b)
- address-computation unit

ISA	TMS320C54xx
precision	0.5
instruction width	16
ops/instruction	1,5
versatility	5.3

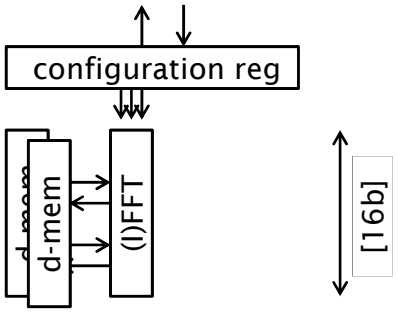
SIMD / EVP-43



- vliw × (simd || scalar)
- simd width = 256 bit
- fixed point + floating point
- 32 scalar + 32 vector registers
- intra-vector (reduction), shuffle, ...

ISA	EVP
precision	0.5
instruction width	80
ops/instruction	31
versatility	1.3

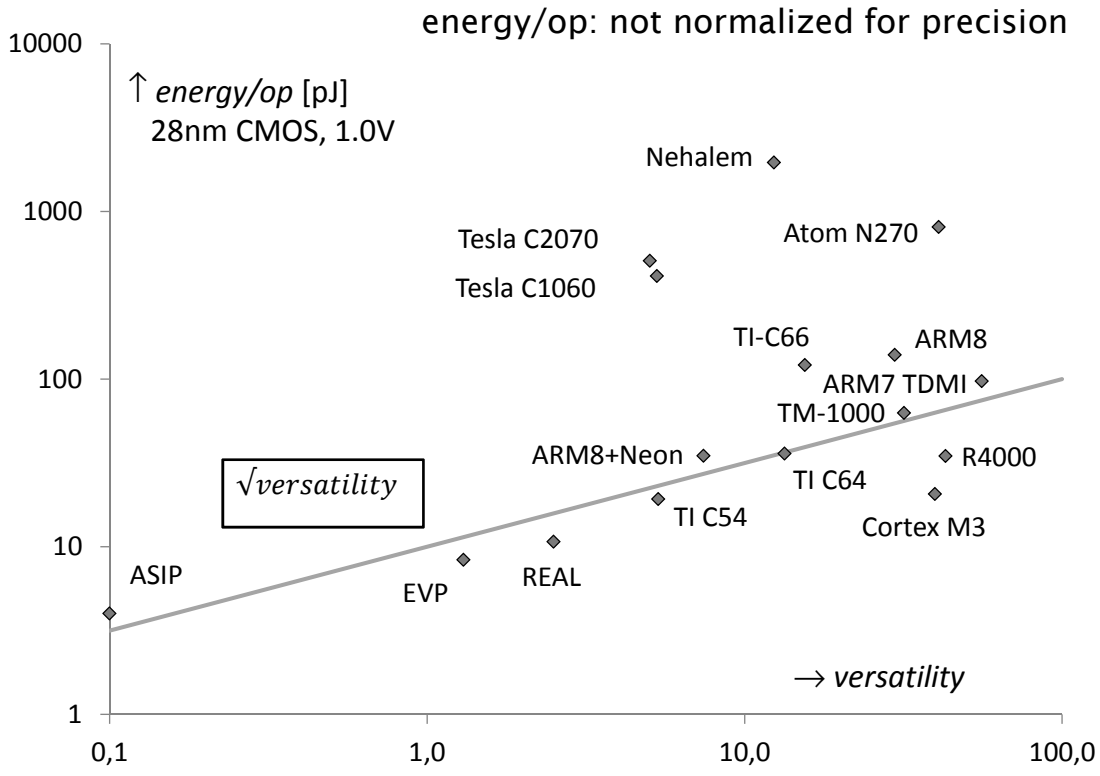
ASIP: (I)FFT (co)processor [UC Davis]



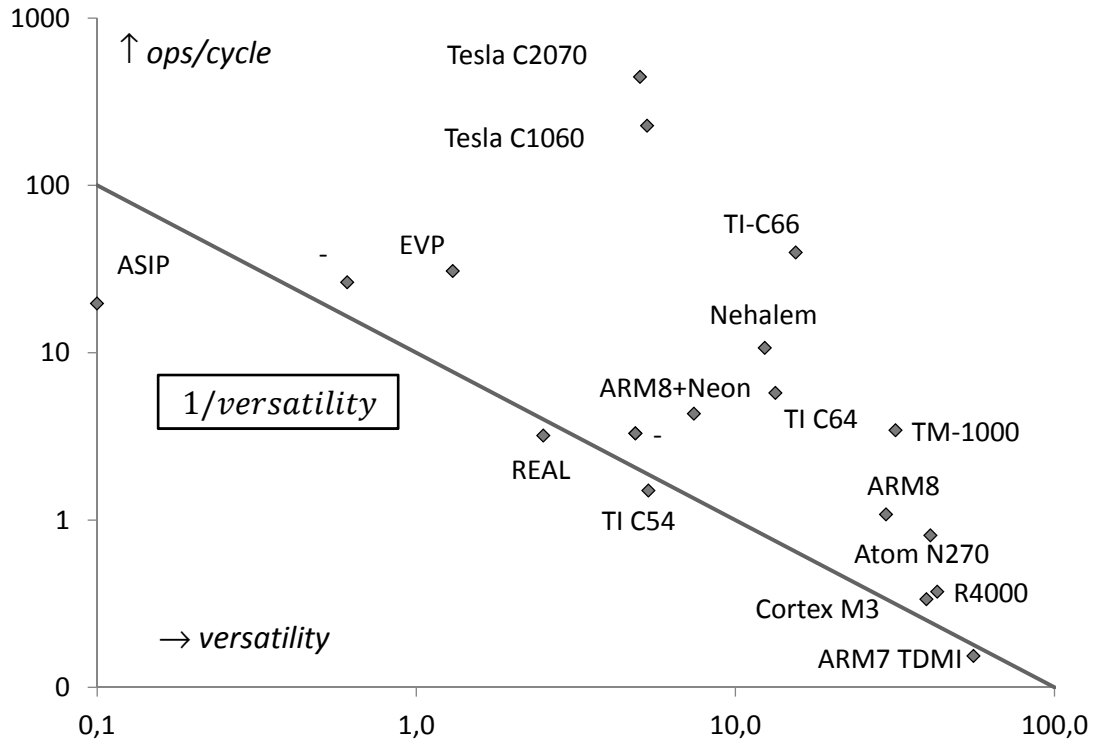
- 16 configuration bits: IFFT vs FFT; block size; pre-processing, block floating point y/n
- 2 local memory banks
- 3 complex multipliers + 12 complex adders

ISA	ASIP
precision	0.5
instruction width	16
ops/instruction	256
versatility	0.1

Energy/operation [pJ/op] versus versatility



Operations/cycle versus versatility



Processor versatility: summary and conclusion

- $versatility = \frac{average\ instruction\ size \times precision}{number\ of\ useful\ operations\ per\ instruction}$
- aligns with qualities like “has many uses or applications”, “is easy to program”, “can change readily from one task to another” , “embraces a variety of algorithms, data types”.
- and orders a range of ISAs consistent with common sense: ASIP ≤ SIMD DSP ≤ GPU ≈ DSP ≤ CPU.
- Empirical evidence suggests a lower bound for energy per operation proportional to $\sqrt{versatility}$

Further questions:

- How to model ISA extensions, e.g. ARM8+Neon *versus* ARM8?
- Versatility of heterogeneous multi-core architectures?
- Area (μm^2 per op/sec) versus versatility?