VLIW PROCESSORS

M. Sonza Reorda

Politecnico di Torino
Dipartimento di Automatica e Informatica

VLIW PROCESSORS

Long Instruction Word **(LIW) and** *Very Long Instruction Word* **(VLIW) processors have long instructions encoding several operations, which are issued in parallel.**

The hardware includes as many functional units, as the operations in a single instruction.

More complex software

It is up to the compiler to decide which instructions to pack together, exploiting parallelism, unrolling loops, scheduling code across basic blocks, etc.

3

Simpler hardware

In a VLIW processor, the hardware does not perform any check on possible dependencies among instructions (this task is completely left to the compiler).

This significantly simplifies the processor.

4

Stalls

When an operation requires stalling (e.g., due to a cache miss) the whole instruction is stalled, in order to preserve the flow decided by the compiler.

Example

Consider a VLIW processor that in every clock cycle can issue:

- two memory references
- two FP operations
- one integer operation or branch.

Which is the processor behavior on the usual loop example?

The branch delay slot is not considered.

Solution

Memory reference 1	Memory reference 2	FP operation 1	FP operation 2	Integer operation/branch
L.D F0,0(R1)	L.D F6,-8(R1)			
L.D F10,-16(R1)	L.D F14,-24(R1)			
L.D F18,-32(R1)	L.D F22,-40(R1)	ADD.D F4,F0,F2	ADD.D F8,F6,F2	
L.D F26,-48(R1)		ADD.D F12,F10,F2	ADD.D F16,F14,F2	
		ADD.D F20,F18,F2	ADD.D F24,F22,F2	
S.D F4,0(R1)	S.D F8,-8(R1)	ADD.D F28,F26,F2		
S.D F12,-16(R1)	S.D F16,-24(R1)			DADDUI R1,R1,#-56
S.D F20,24(R1)	S.D F24,16(R1)			
S.D F28,8(R1)				BNE R1,R2,Loop

7

Solution

Memory reference 1	Memory reference 2
L.D F0,0(R1)	L.D F6
L.D F10,-16(R1)	L.D
L.D F18,-32(R1)	L.D F22,-40
L.D F26,-48(R1)	
S.D F4,0(R1)	S.D F8,-8(R1)
S.D F12,-16(R1)	S.D F16,-24(
S.D F20,24(R1)	S.D F24,16(R1
S.D F28,8(R1)	

The code containing 7 iterations is executed in 9 clock cycles, corresponding to 1.29 cycles per vector element.

In the average, 2.5 operations per clock cycle are executed.

The efficiency (percentage of available slots containing an operation) is about 60%.

The number of required FP registers is 8, compared with 2 for the normal pipelined processor, 5 for the unrolled version, and 6 for the superscalar one.

8

Instruction size

In our example we have 5 functional units.

To control each functional unit we need 16 to 24 bits.

Therefore, the length of an instruction is from 112 to 168 bits.

9

Advantages

A VLIW processor does not require the hardware for choosing the instructions to be executed in parallel (i.e., for detecting dependencies).

LIMITATIONS

The performance that can be attained by a multiple-issue processor is limited by

- **inherent limitations of ILP in programs**
- **difficulties in building the hardware**
- **limitations specific to a superscalar or VLIW processor.**

Limitations in the inherent ILP

It is difficult to find a sufficient number of independent instructions that can be executed in parallel.

This is even more difficult due to pipelined functional units, having a latency greater than 1.

In general, to avoid any stall we need to find a number of independent operations roughly equal to

average pipeline depth ´ number of functional units.

With 5 functional units, and an average pipeline depth of 4 clock cycles, we need as many as 20 independent operations to avoid stalls.

Hardware cost

By increasing the number of functional units, there is also an increase in the bandwidth towards the register file and the memory.

This means increasing the hardware complexity and possibly decreasing the performance.

Possible solutions are

- memory interleaving
- multiport memories
- multiple access per clock cycle memories.

Hardware Cost (II)

It varies between the maximum cost of fully dynamically scheduling processors and the minimum cost of VLIW processors.

Most current processors are based on a combination of static and dynamic scheduling.

Code size

It is much larger for VLIW processors, mainly due to two factors:

- loops are intensively unrolled to extract more parallelism
- the empty slots in instruction encoding.

Sometimes instructions are compressed in memory and expanded when loaded to the processor.

Memory access

In VLIW processors, access to memory is often a bottleneck, since the required bandwidth is higher, and stalls due to cache misses cause a stall on the whole processor (all functional units are synchronized by the execution order decided by the compiler).

Binary Code Compatibility

Any change in the implementation of a VLIW processor (e.g., changing the latency of a functional unit) requires recompiling the code.

This is a major disadvantage with respect to superscalar processors, which can easily be made binary compatible with their previous versions.

Object code translation or emulation will possibly solve this problem in the future.

17

Examples

- **Trimedia TM32**
- **Transmeta Crusoe.**

18

Trimedia TM32

It is intended for media applications (e.g., MPEG compression and decompression).

Every instruction contains five operations.

The processor

- **is completely statically scheduled**
- **does not detect hazards.**

19

Transmeta Crusoe

It is intended for the low-power market (mobile PCs and mobile Internet appliances).

It guarantees instruction set compatibility with the x86 instruction set (a software system translates from the x86 instruction set to the Crusoe one).

Instructions come in two sizes:

- **64 bits (2 operations)**
- **128 bits (4 operations).**

The processor includes:

- **a 6-stage pipeline for integer instructions**
- **a 10-stages pipeline for FP instructions.**

20