Computer Architecture
Lecture 18

Very- Long Instruction Word (VLIW) Computer Architecture

Very Long Instruction Word (VLIW)

- Instruction level parallelism (ILP): multiple instructions fetched and executed simultaneously.
- ILP is used in addition to pipelining.
- Processors with ILP are called *multiple-issue processors* – multiple instructions launched in 1 clock cycle. Two ways:
  - MIMD: Multiple Instructions Multiple Data
    - Superpipeline
    - Superscalar – dynamic multiple issue
    - Very long instruction word (VLIW) – static multiple issue
  - SIMD: Single Instruction Multiple Data
    - Vector processor

The method for exploiting parallelism

- The key to higher performance in microprocessors for a broad range of applications is the ability to exploit fine-grain, instruction-level parallelism:

  + pipelining

  + multiple processors

  + superscalar implementation

  + specifying multiple independent operations per instruction

Multiple Issue

Multiple instructions can be issued by

- **Statically** using compiler (at compile time), technique is called Static Multiple Issue
  - VLIW (Very long instruction word) processor
- **Dynamically** (during execution using hardware) technique called Dynamic Multiple issue
  - Superscalar processor

Data and Control Hazards

- In static issue processors, some or all of the data and control hazards are handled statically by the compiler.

- In dynamic issue processors most of the data and control hazards are handled using hardware techniques.

Instruction issue slots?

- How does the processor determine how many instructions and which instructions can be issued in a given clock cycle?
- In most static issue processors, this process is at least partially handled by the compiler
- In dynamic issue designs, it is normally dealt with at runtime by the processor.

Static Multiple issue

- Static Multiple issue processors use compiler to assist with packing instructions and handling hazards.
- Set of instructions which are issued in one clock cycle called **issue packet**.
- Issue packet can also be interpreted as one large instruction with multiple operations.
- The **issue packet** determined statically by the compiler leads to approach **Very Long Instruction Word** (VLIW)

VLIW

- Multiple independent instructions are issued in one clock cycle
- Compiler finds out independent instructions and issues them accordingly.
- Compiler's responsibilities include static branch prediction and code re-ordering to avoid all hazards.
- Intel IA-64 uses this approach called Explicitly Parallel Instruction Computer (EPIC)
  - Itanium processor, available since (2000)
  - Itanium 2 processor, available since (2002)

Two issue MIPS

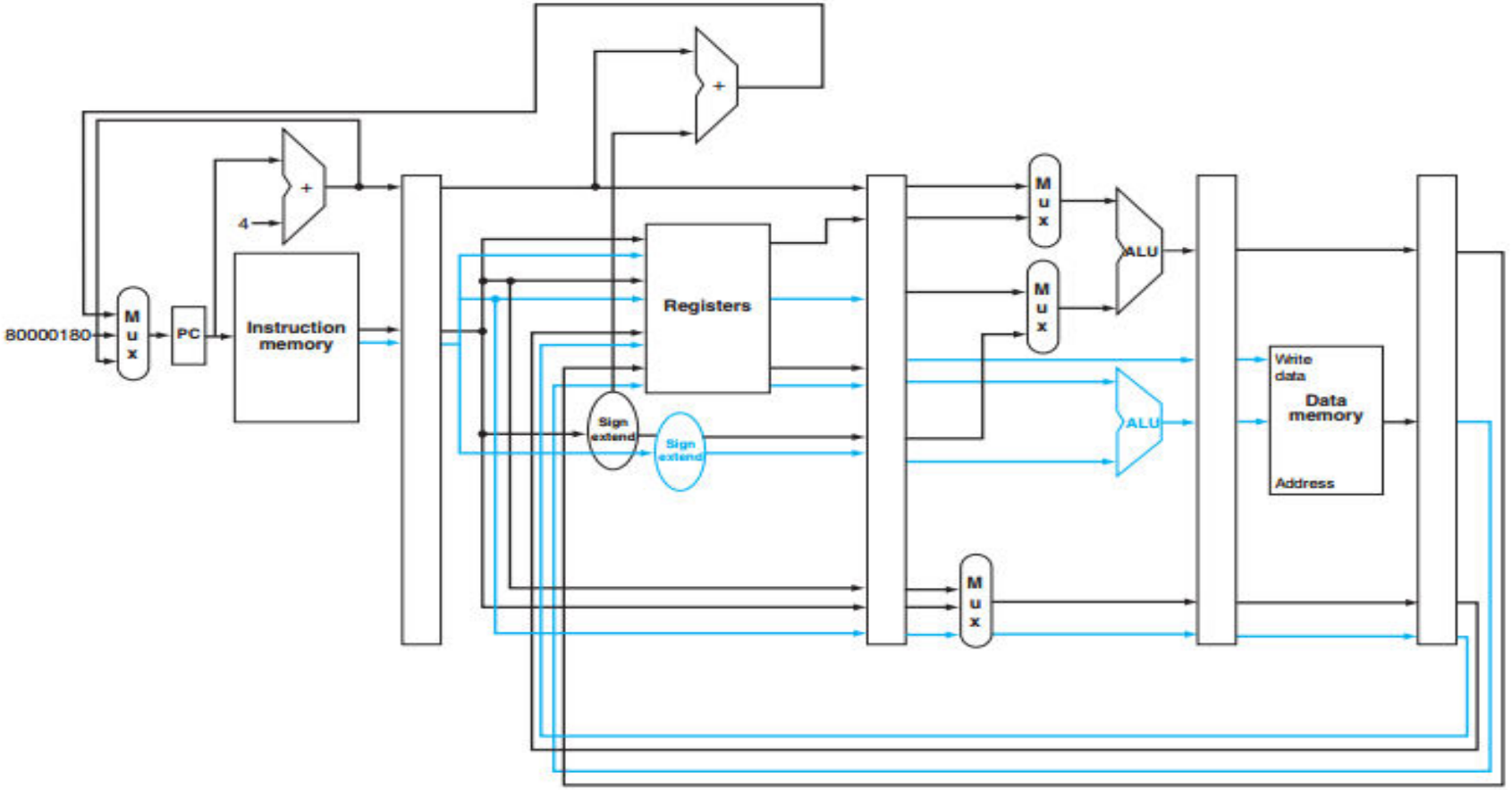
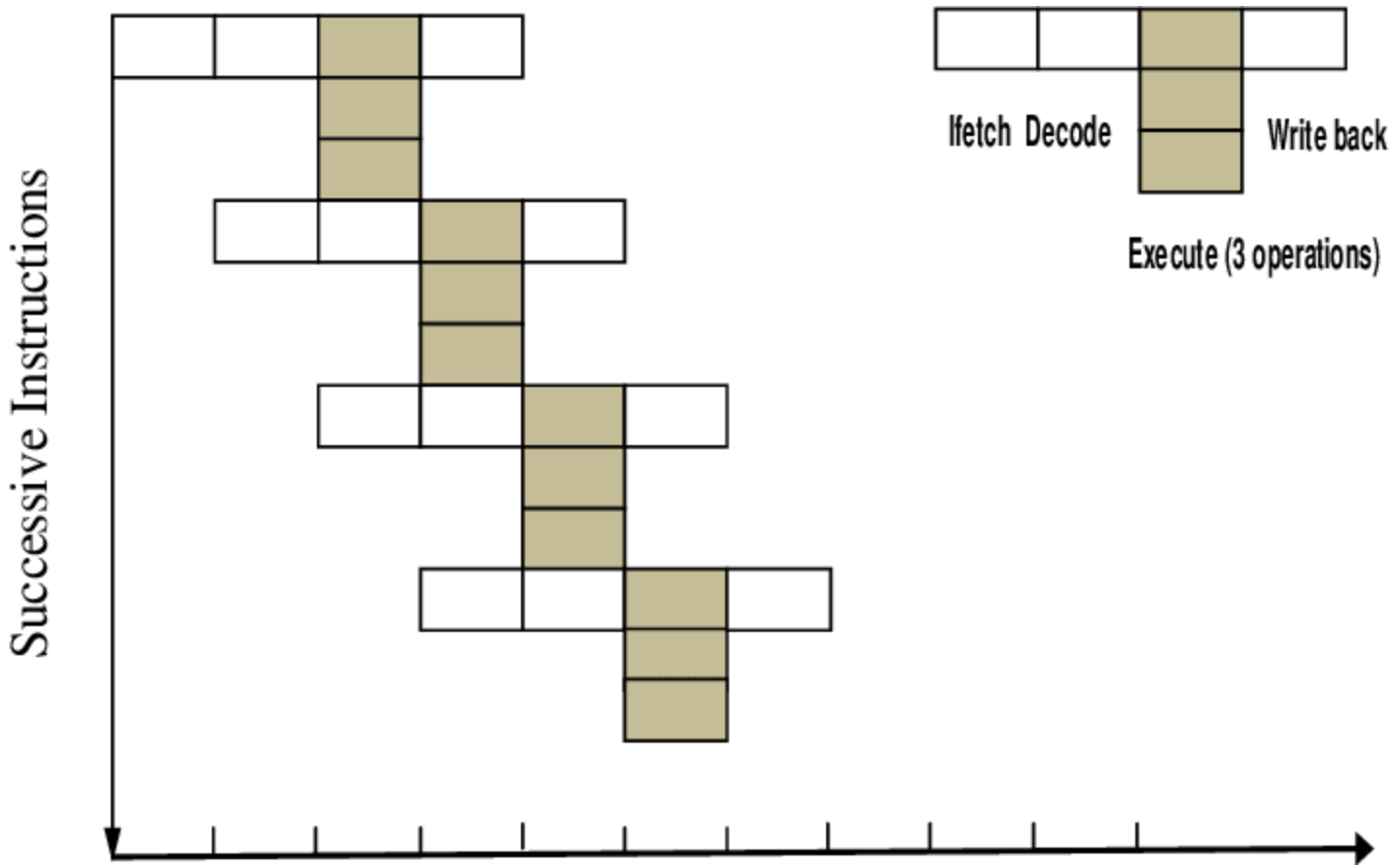
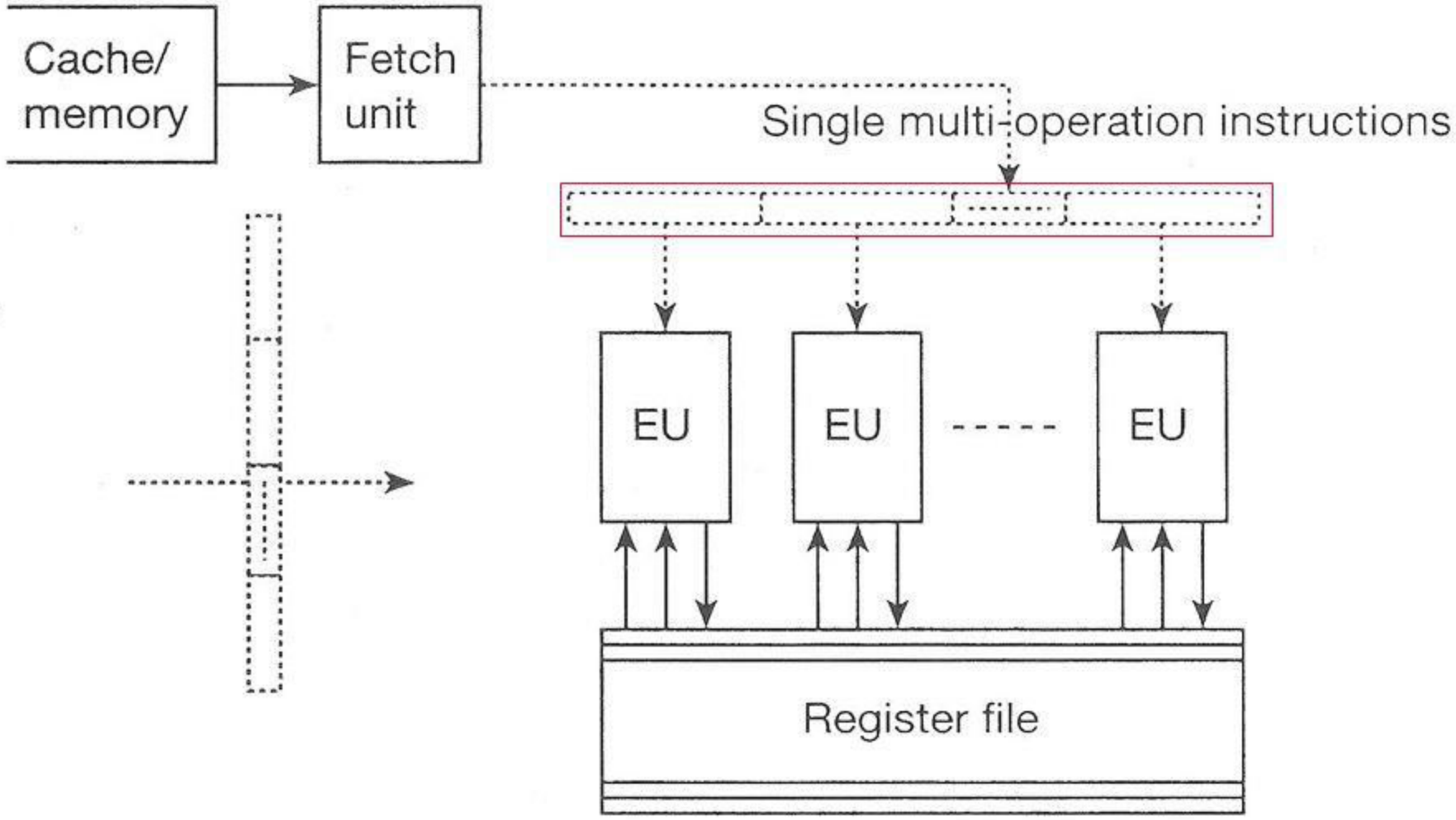


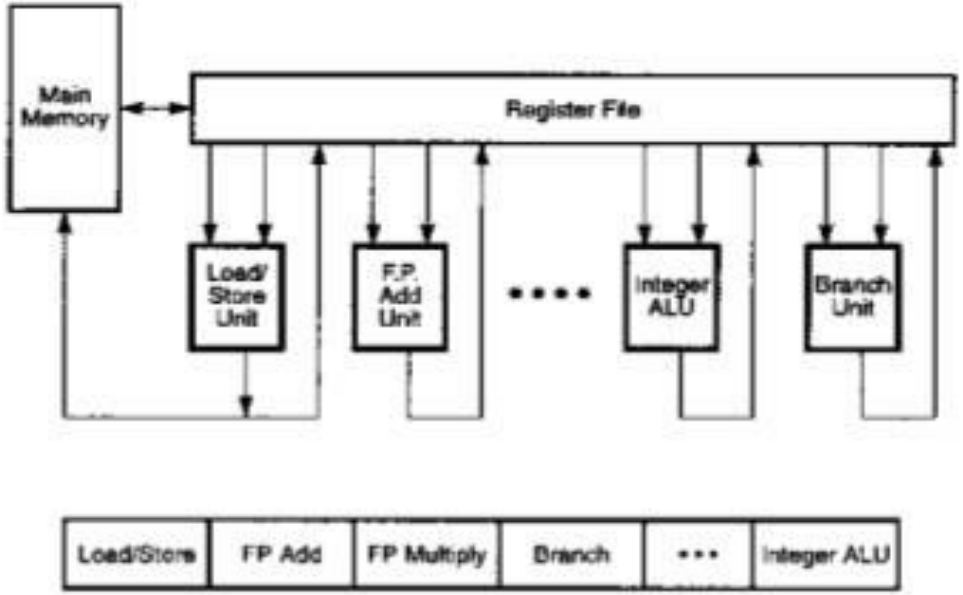
FIGURE 6.45 A static two-issue datapath. The additions needed for double issue are highlighted: another 32 bits from instruction memory, two more read ports and one more write port on the register file, and another ALU. Assume the bottom ALU handles address calculations for data transfers and the top ALU handles everything else.


Successive Instructions
Ifetch
Decode
Execute (3 operations)
Write back

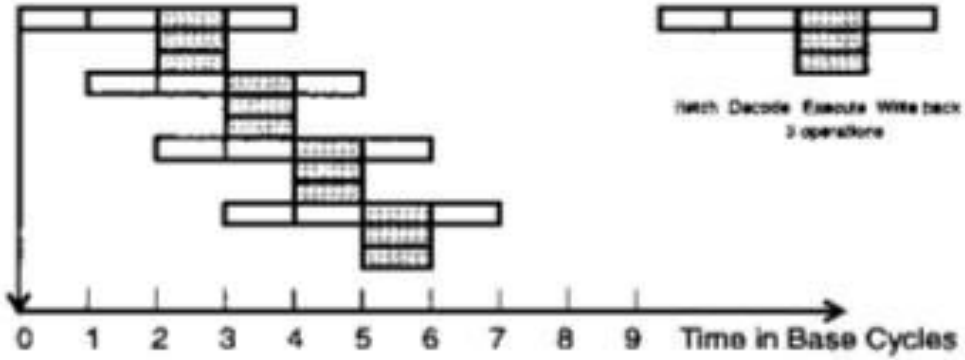
Basic VLIW Approach



VLIW approach



(a) A typical VLIW processor and instruction format



(b) VLIW execution with degree $m = 3$

Figure The architecture of a very long instruction word (VLIW) processor and its pipeline operations. (Courtesy of Multiflow Computer, Inc., 1987)

VLIW characteristics

- VLIW contains multiple primitive instructions that can be executed in parallel by functional units of a processor.
- The compiler packs a number of primitive, non-interdependent instructions into a very long instruction word
- Since multiple instructions are packed in one instruction word, the instruction words are much larger than CISC and RISC's.

The VLIW compiler

- The compiler specifies the primitive instructions per VLIW instruction word.
- The compiler must guarantee that the multiple primitive instructions which group together are independent so they can be executable in parallel.

VLIW principles

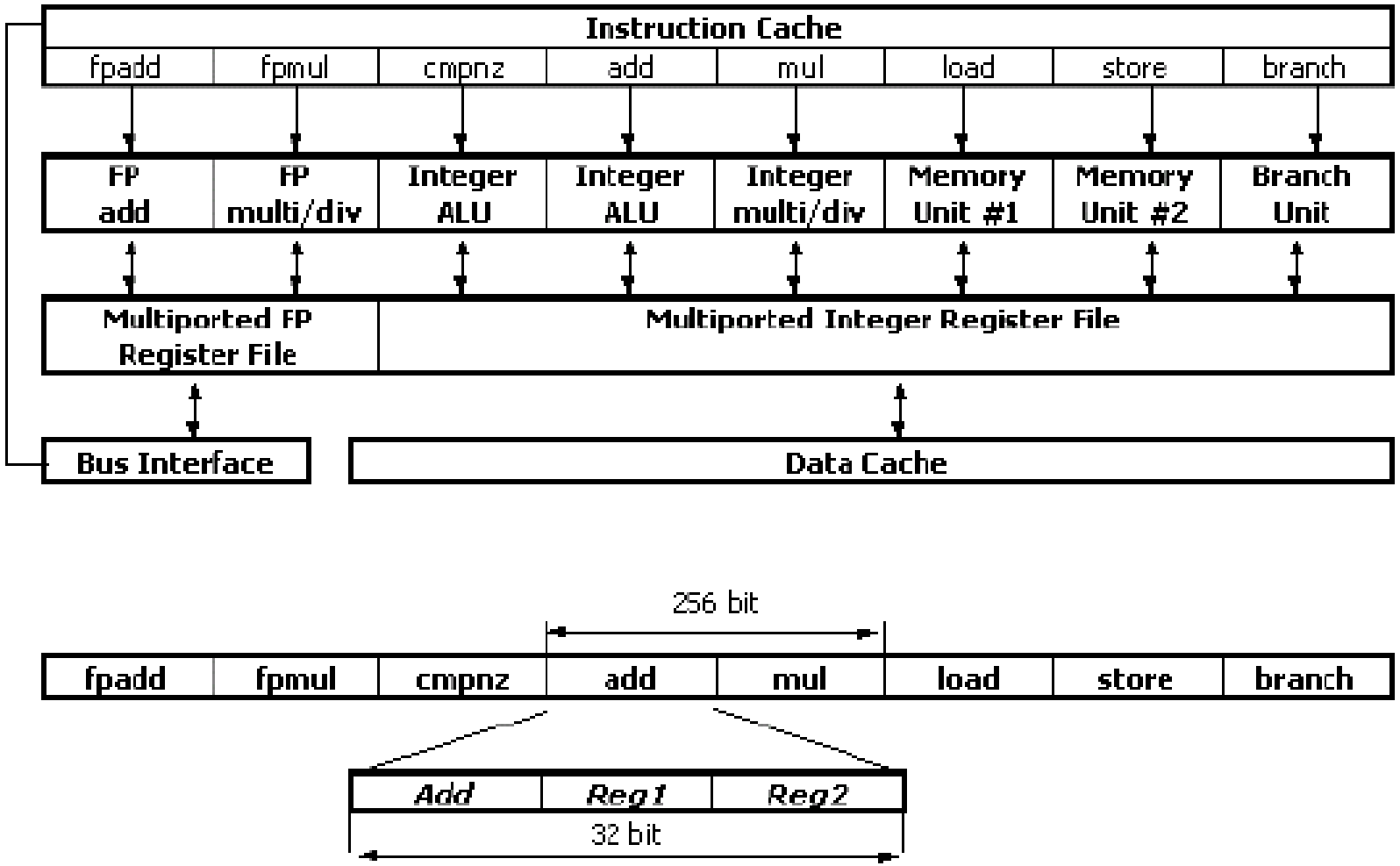
1. The compiler analyzes dependence of all instructions among sequential code, tries to extract as much parallelism as possible.
2. Based on the analysis, the compiler re-codes the piece of sequential code in VLIW instruction words.
3. The work left with VLIW hardware is only fetch the VLIWs from cache, decode them, and then dispatch the independent primitive instructions to corresponding function units and execute.

4. One VLIW instruction word contains maximum 8 primitive instructions.
5. Each time, one VLIW instruction word is fetched from cache and decoded.
6. After decoding, all primitive instructions in this VLIW word are issued to functional units in parallel for execution.
7. These primitive instructions are from the same VLIW word, so they are guaranteed to be independent.

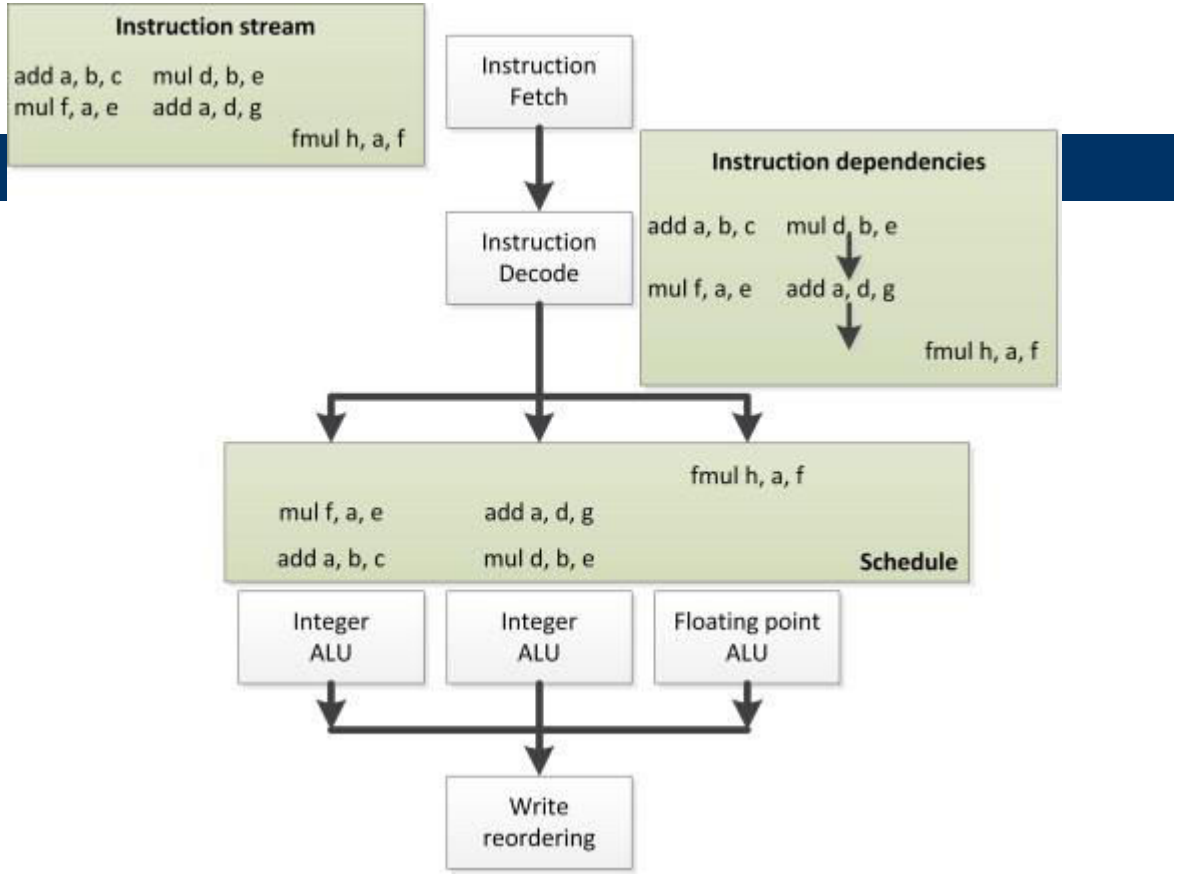
SOFTWARE INSTEAD OF HARDWARE: IMPLEMENTATION ADVANTAGES OF VLIW

VLIW instructions explicitly specify several independent operations— decode the instruction and dispatch hardware that tries to reconstruct parallelism from a serial instruction stream. The processor does not need to consider whether or not the instructions are parallel.

Generating of VLIW instruction words



A hypothetical VLIW processor architecture


Instruction stream
add a, b, c
mul d, b, e
mul f, a, e
add a, d, g
fmul h, a, f
Instruction Fetch
Instruction Decode
Instruction dependencies
add a, b, c
mul d, b, e
mul f, a, e
add a, d, g
fmul h, a, f
fmul h, a, f
mul f, a, e
add a, d, g
add a, b, c
mul d, b, e
Schedule
Integer ALU
Integer ALU
Floating point ALU
Write reordering

CISC (Complex Instruction Set Computing)

CISC (Complex Instruction Set Computing)

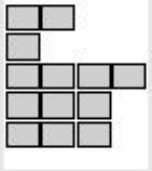
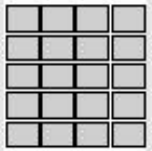
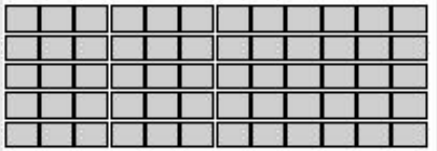
- instructions are quite complex and have variable length.
- a relatively small number of registers, and are capable of accessing memory locations directly.
- Complex instructions are sequenced in microcode in modern CISC processors.

RISC(Reduced Instruction Set Computing)

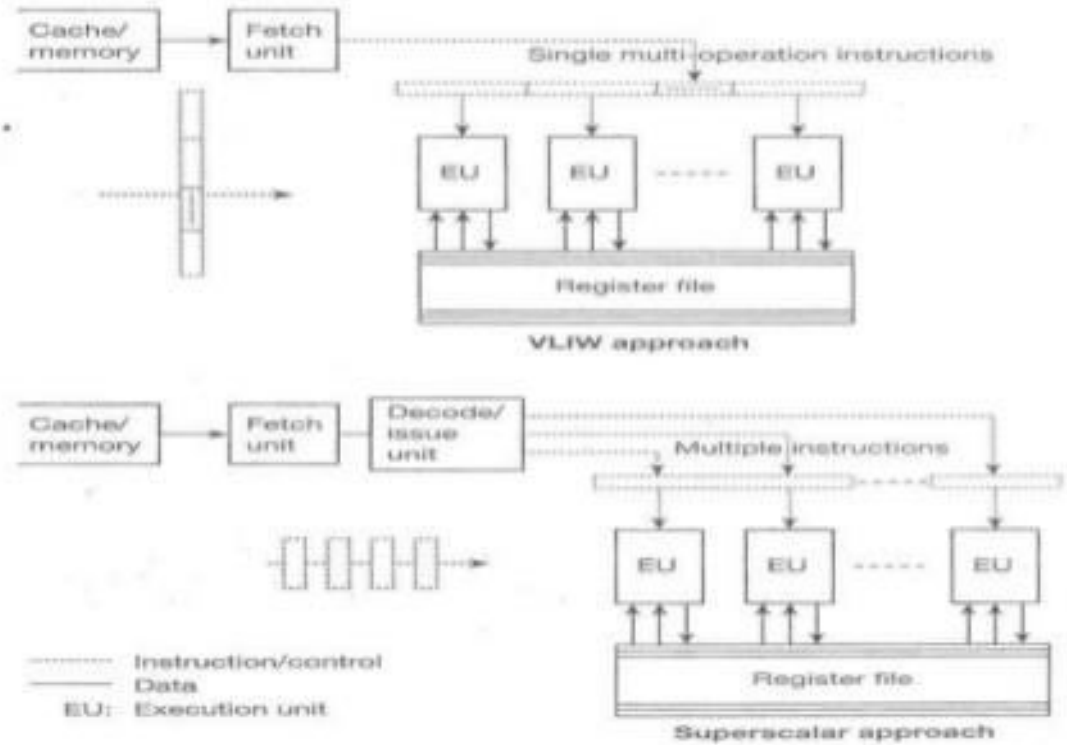
RISC(Reduced Instruction Set Computing)

- instructions are of fixed length and of a regular format.
- Operations are performed on registers only, of which a larger number is available than on CISC processors. The only memory operations are load and store.
- The hardware in RISC processors is simpler because the RISC architecture relies more on the compiler for sequencing complex operations.

Comparison of VLIW, CISC,RISC

ARCHITECTURE CHARACTERISTIC	CISC	RISC	VLIW
INSTRUCTION SIZE	Varies	One size, usually 32 bits	One size
INSTRUCTION FO RMAT	Field placement varies	Regular, consistent placement of fields	Regular, consistent placement of fields
INSTRUCTION SEMANTICS	Varies from simple to complex; possibly many dependent operations per instruction	Almost always one simple operation	Many simple, independent operations
REGISTERS	Few, sometimes special	Many, general-purpose	Many, general-purpose
MEMORY REFE RENCES	Bundled with operations in many different types of instructions	Not bundled with operations, i.e., load/store architecture	Not bundled with operations, i.e., load/store architecture
HARDWARE DESIGN FOCUS	Exploit microcoded implementations	Exploit implementations with one pipeline and & no microcode	Exploit implementations with multiple pipelines, no microcode & no complex dispatch logic
PICTURE OF FIVE TYPICAL INSTRU CTIONS = 1 BYTE			

Difference Between VLIW & Superscalar Architecture



Conclusion

1. The highly parallel implementation is much simpler and cheaper than its counterparts.
2. The encoding of VLIW words implies parallelism among their primitive instructions, which results in reduced hardware complexity.
3. The complier must assemble multiple primitive instructions into a single VLIW, to make sure that multiple function units are kept busy.

Conclusion(cont.)

4. The compiler optimizes software pipeline; by re-ordering tries to find the most parallelism in the sequential code.
5. The microprocessor performance is dependent on how the compiler produces VLIW words.