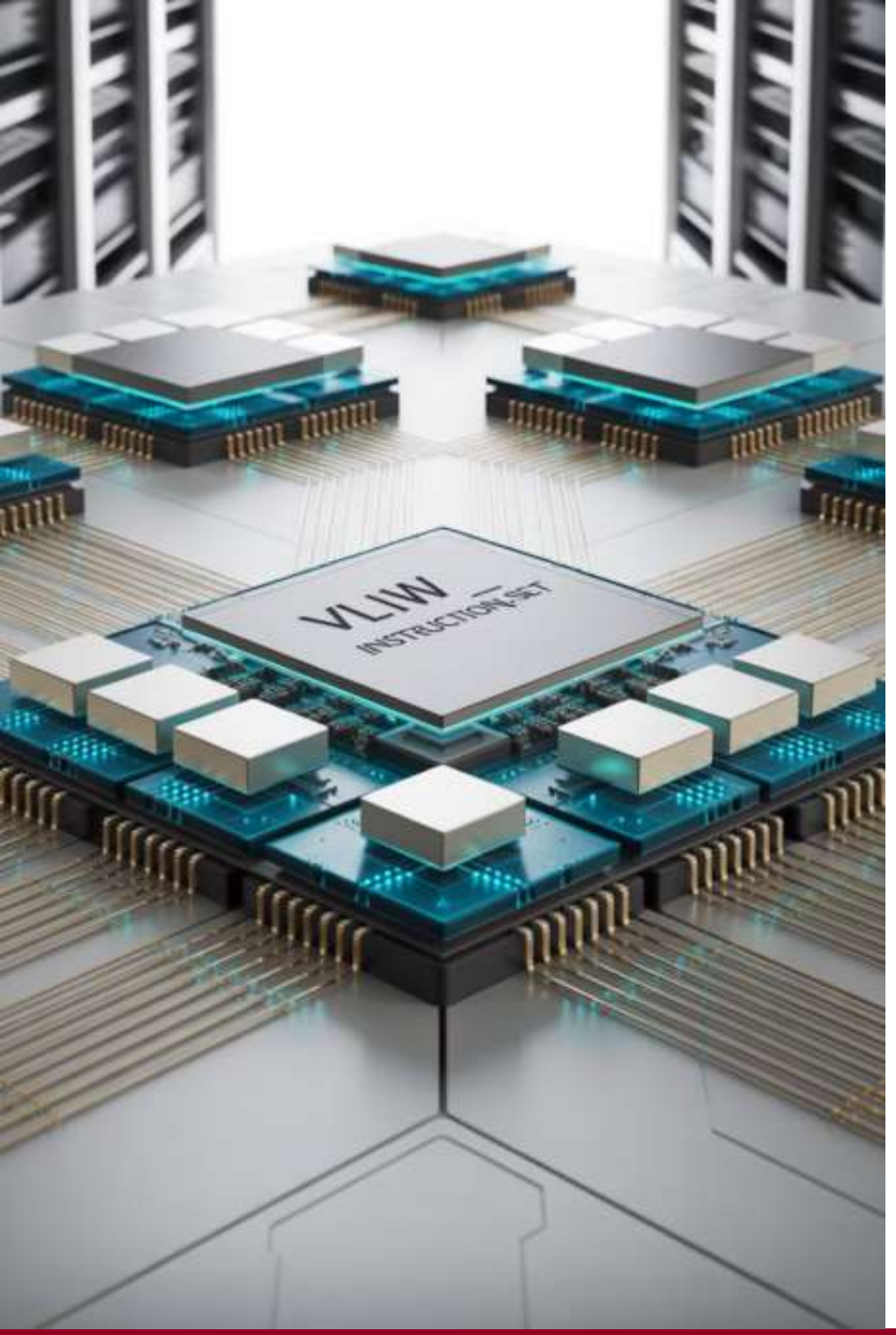
Multiple-Issue Approaches

Approach	Issue Structure	Hazard Detection	Scheduling	Examples
Superscalar (static)	Dynamic	Hardware	Static	MIPS, ARM Cortex-A8
Superscalar (speculative)	Dynamic	Hardware	Dynamic with speculation	Intel Core i3/i5/i7, AMD Phenom, IBM Power 7
VLIW/LIW	Static	Primarily software	Static	TI C6x (signal processing)

Each approach represents different trade-offs between hardware complexity and compiler responsibility for extracting parallelism.



VLIW Architecture Fundamentals

Key Characteristics

- Multiple independent functional units
- Operations packaged into one very long instruction
- Typical width: 5+ operations per instruction
- 80-120 bits per instruction
- Itanium: 6 operations per instruction packet

Parallelism Extraction

- Loop unrolling to generate straight-line code
- Local scheduling within basic blocks
- Global scheduling across branches (more complex)
- Trace scheduling (specialized for VLIWs)

Example

```
Loop: L.D    F0,0(R1)    ;F0=array element
      ADD.D  F4,F0,F2    ;add scalar in F2
      S.D    F4,0(R1)    ;store result
      DADDUI R1,R1,#-8   ;decrement pointer
      BNE    R1,R2,Loop  ;branch R1!=R2
```

Unroll the loop

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

Two memory references
Two FP units
One integer operation

Per cycle

Multiple-Issue vs. Vector Processing

- When parallelism comes from simple FP loop unrolling, vector processors may be equally or more efficient than multiple-issue processors.
- Multiple-issue processors excel at:
  - Extracting parallelism from less structured code
  - Easily caching all forms of data
- For these reasons, multiple-issue approaches have become the primary method for exploiting instruction-level parallelism, with vector capabilities often added as extensions.