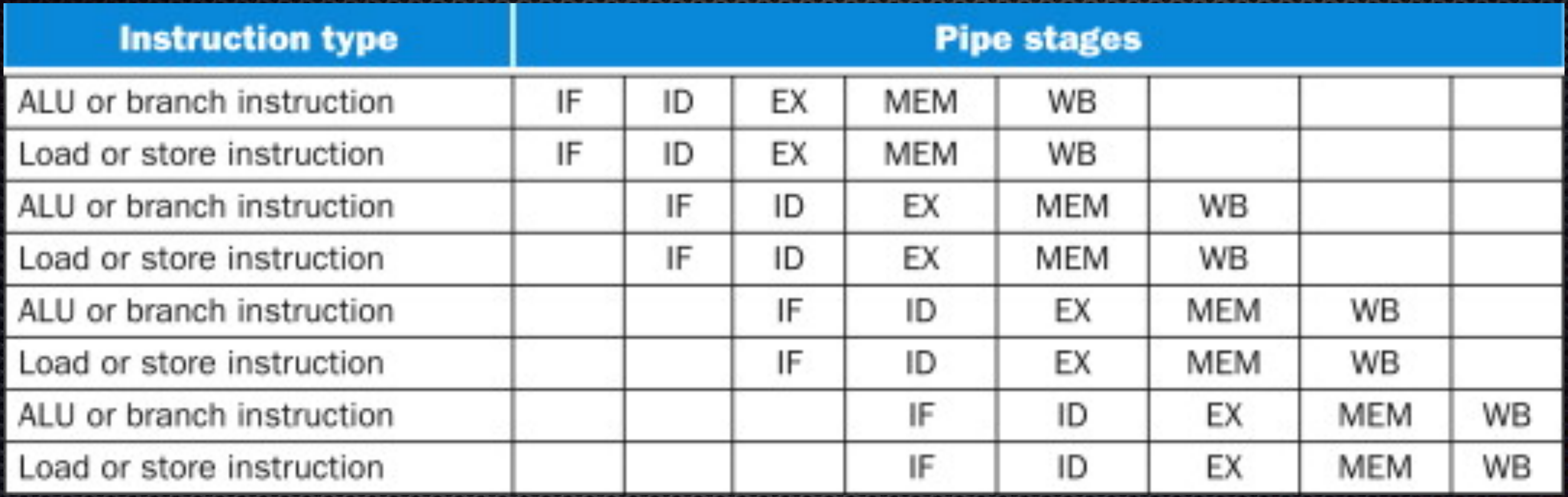
Parallelism In Pipelining

- Static Multiple Issue
  - Chosen at compile time
- Dynamic Multiple Issue
  - Chosen during running

Static Multiple Issue Pipeline



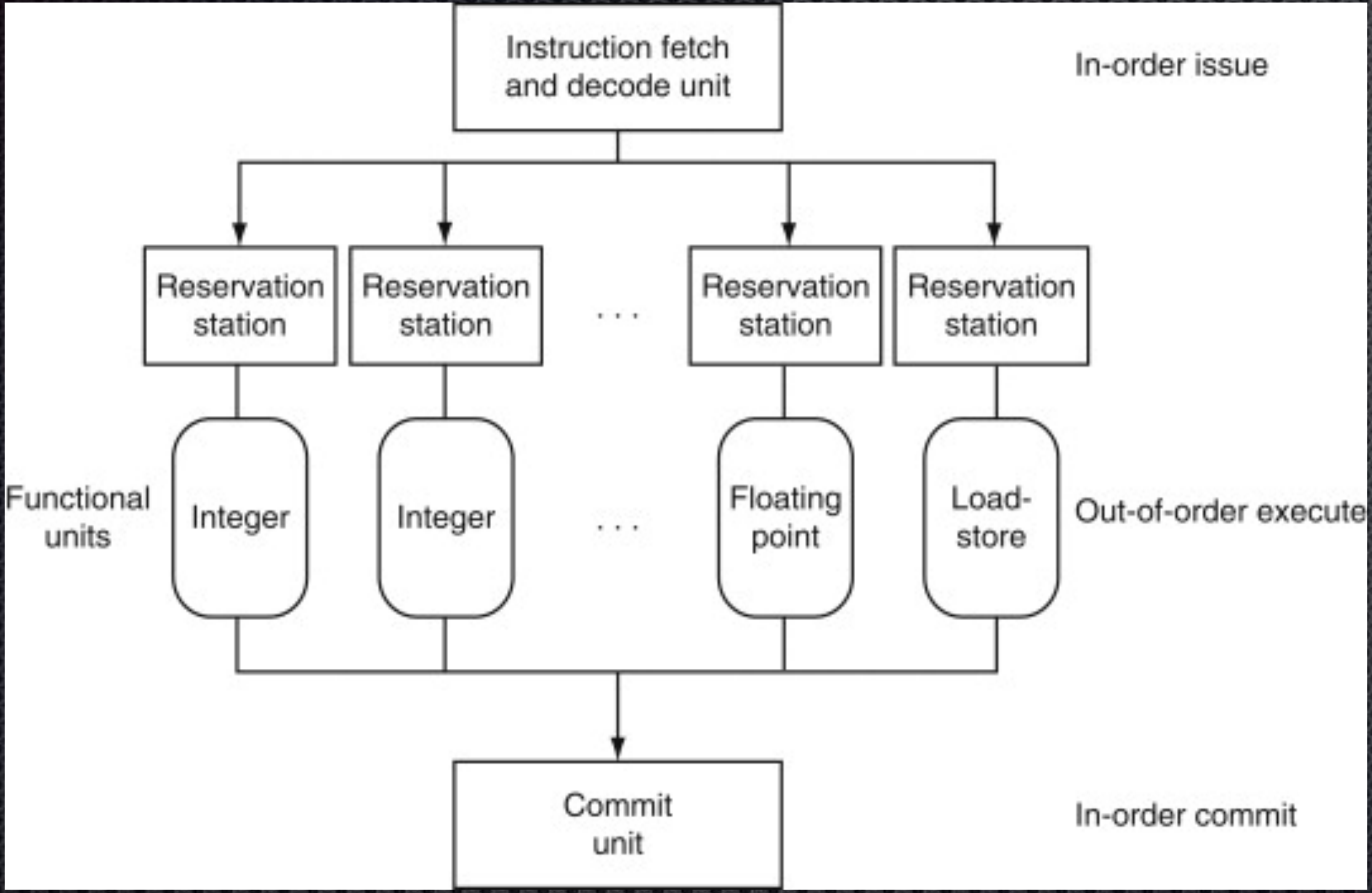
Instruction type	Pipe stages							
ALU or branch instruction	IF	ID	EX	MEM	WB			
Load or store instruction	IF	ID	EX	MEM	WB			
ALU or branch instruction		IF	ID	EX	MEM	WB		
Load or store instruction		IF	ID	EX	MEM	WB		
ALU or branch instruction			IF	ID	EX	MEM	WB	
Load or store instruction			IF	ID	EX	MEM	WB	
ALU or branch instruction				IF	ID	EX	MEM	WB
Load or store instruction				IF	ID	EX	MEM	WB

```
add x2, x3, x4
ldur x9, [x0, #4]
add x5, x2, x2
stur x2, [x0, #8]
sub x7, x5, x2
cbz x2, #3
.
..
...
```

Specially chosen to avoid hazards….

Instruction	Cycle1	Cycle2	Cycle3	Cycle4	Cycle5	Cycle6	Cycle7	Cycle8
add x2, x3, x4	IF	ID	EX	MEM	WB			
ldur x9, [x0, #4]	IF	ID	EX	MEM	WB			
add x5, x2, x2		IF	ID	EX	MEM	WB		
stur x2, [x0, #8]		IF	ID	EX	MEM	WB		
sub x7, x5, x2			IF	ID	EX	MEM	WB	
cbz x2, #3				IF	ID	EX	MEM	WB

Dynamic Multiple Issue

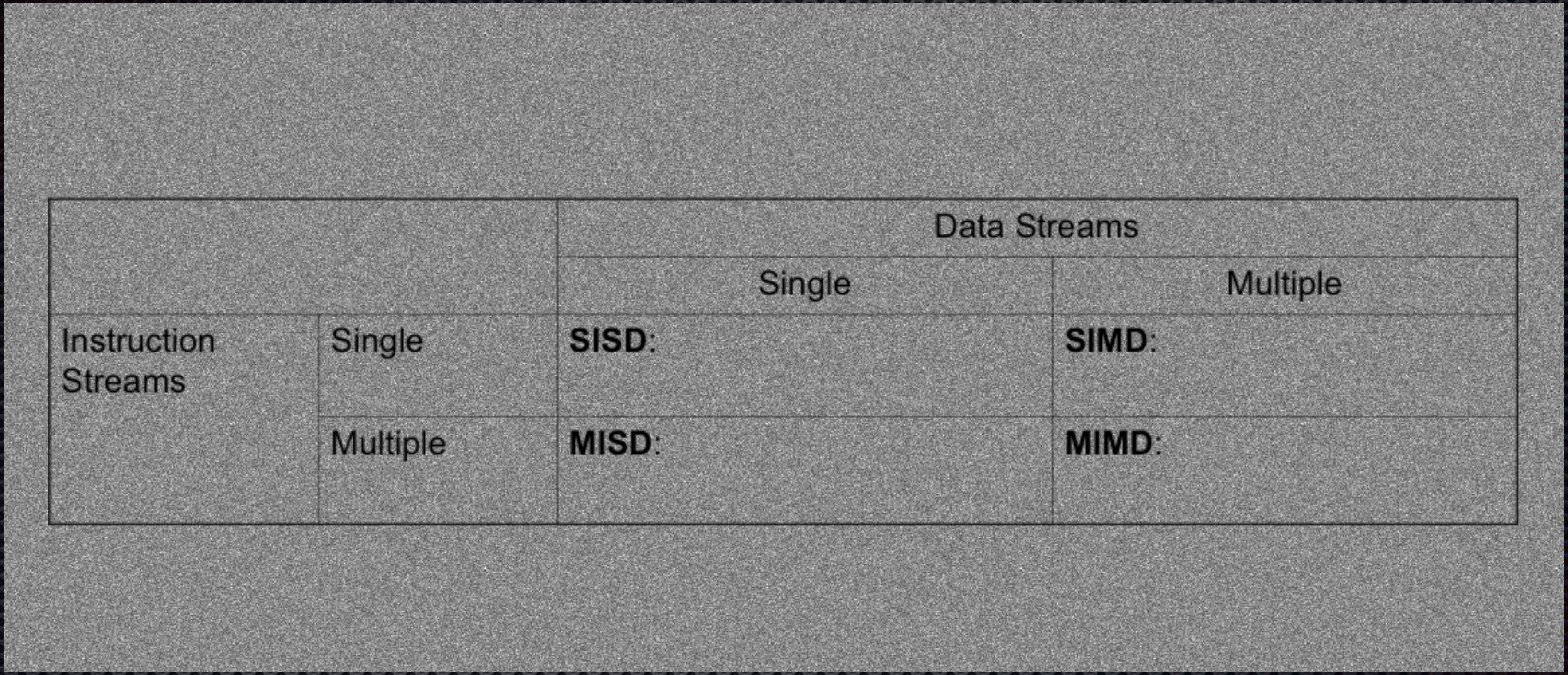


```
add x2, x3, x4
ldur x9, [x0, #4]
add x5, x2, x2
stur x2, [x0, #8]
sub x7, x5, x2
cbz x2, #3
.
..
…
```

Parallel Processing

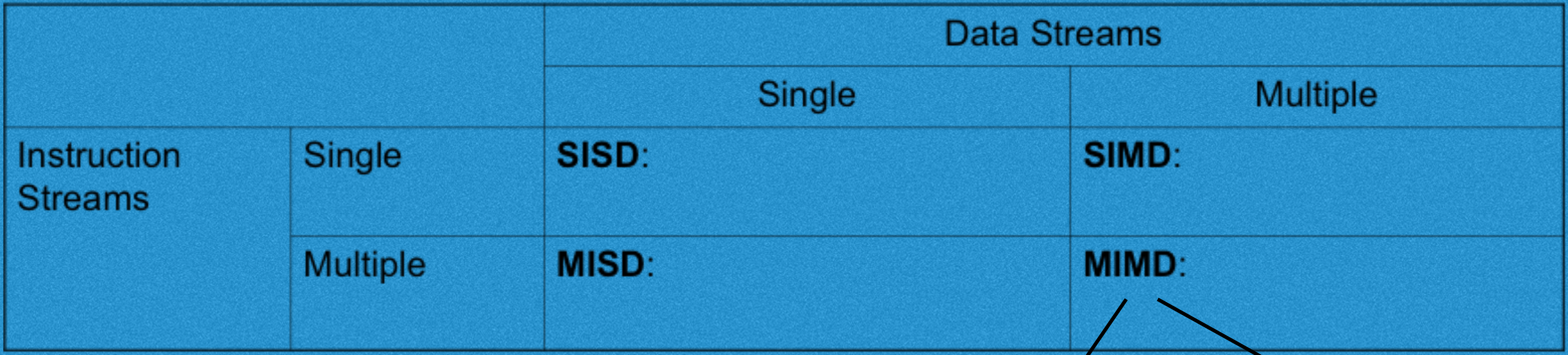
- Need to get significant performance improvement
  - Otherwise just get a faster uniprocessor
- Difficulties
  - Partitioning
  - Coordination
  - Communications overhead

Flynn’s Taxonomy



		Data Streams	
		Single	Multiple
Instruction Streams	Single	**SISD**:	**SIMD**:
	Multiple	**MISD**:	**MIMD**:

Flynn’s Taxonomy



		Data Streams	
		Single	Multiple
Instruction Streams	Single	**SISD**:	**SIMD**:
	Multiple	**MISD**:	**MIMD**:

Shared Memory Processing

Multicore
Threading
GPUs

Single Program Multiple Data

Grid computing
Clusters

Amdahl’s Law

- $T_{improved} = T_{parallelizable}/Factor + T_{sequential}$
- $Speedup = T_{old}/T_{improved} =$ $(T_{parallelizable} + T_{sequential})/(T_{parallelizable}/Factor + T_{sequential})$
- $F_{parallelizable} = T_{parallelizable} /(T_{parallelizable} + T_{sequential})$
- $1\text{-}F_{parallelizable} = T_{sequential} /(T_{parallelizable} + T_{sequential})$
- $Speedup = 1/(F_{parallelizable}/Factor + (1\text{-}F_{parallelizable}))$

Speedup

A shared memory processing system has 4 processors. A program consists of a section that must be executed sequentially and requires 10 ns. The remainder of the program can be parallelized and requires 90 ns. What is the speedup obtained when running this program on 4 processors?

Speedup

A shared memory processing system has 4 processors. A program consists of a section that must be executed sequentially and requires 10 ns. The remainder of the program can be parallelized and requires 90 ns. What is the speedup obtained when running this program on 4 processors?

Non parallel processing time:

100 ns

Parallel processing time:
10 + 90/4 = 32.5 ns

Speedup = 100/32.5 = 3.08

Amdahl's Law

If I have 10 processors, what fraction of a program must be parallelizable in order to get a speedup of 5?

$5 = 1/(F/10 + (1-F))$

$5*(F/10 + 1-F) = 1$

$F/2 + 5*(1-F) = 1$ multiply by 2
$F+ 10 - 10F = 2$
$8 = 9F$

$F = 8/9 = 0.889$