# Topic V2E

Speculation

Reading: (Section 4.10)

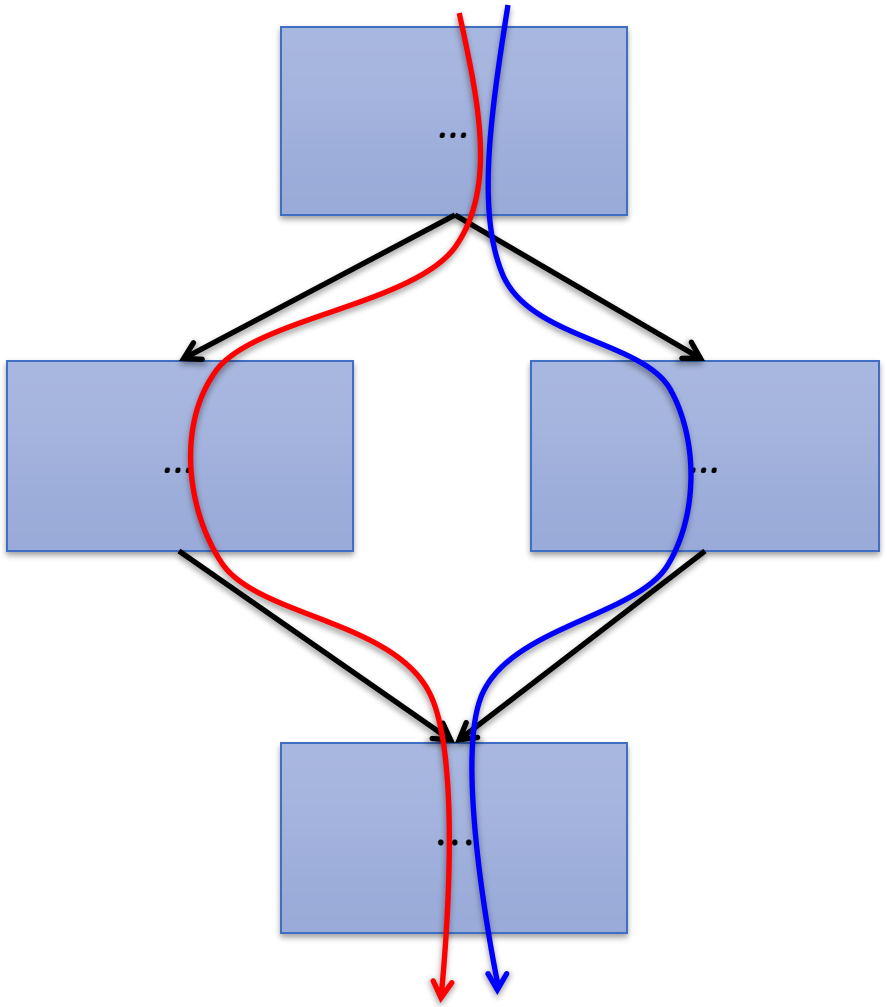
# Speculation

- “Guess” what to do with an instruction
  - Start operation as soon as possible
  - Check whether guess was right
    - If so, complete the operation
    - If not, roll-back and do the right thing
- Common to static and dynamic multiple issue

# Speculation - Examples

Speculate on branch outcome

- Roll back if path taken is different



# Speculation - Examples

Speculate on load

Roll back if location is updated

```
 lw s2, 0(s3)          # try to execute load before store

sw t0, 0(t1)
…
lw s2, 0(s3)
```

If store was to same memory location
as load, processor has to issue load again.

# Compiler Speculation

Compiler can reorder instructions

e.g., move load before store

Include “fix-up” instructions to recover from incorrect guess

```
…
sw t0, 0(t1)
…
lw s2, 0(s3)
…
add s3, s4, s2
```

```
    lw s2, 0(s3)
    add s7, s4, s2
    …
    sw t0, 0(t1)
    …
    bne t1, s3, skipFixUp
    lw s2, 0(s3)
    add s7, s4, s2
skipFixUp
```

Speculate that load and store are to different locations.

Redo load if incorrect.

# Hardware Speculation

- Hardware can look ahead for instructions to execute
  - Buffer results until it determines that they are needed
  - Flush buffers on incorrect speculation

# Speculation and Exceptions

What if exception occurs on a speculatively executed instruction?

- e.g., speculative load before null-pointer check

Static speculation

- Can add ISA support for deferring exceptions

Dynamic speculation

- Can buffer exceptions until instruction completion (which may not occur)

# Speculation

- Predict branch and continue issuing
  - Do not commit until branch outcome is determined
- Load speculation
  - Avoid load and cache miss delay
    - Predict the effective address
    - Predict loaded value
    - Load before completing outstanding stores
    - Bypass stored values to load unit
  - Don’t commit load until speculation cleared