

Data Dependency Graph & Software Pipelining

- Jongtae Park -

Embedded Systems Language &
compiler lab



Organization

- Dependences between BB or Instructions
- Data Dependence Graph
- the basic concept of Software pipelining
- Q & A

Dependence

- Data dependence
- Name dependence
 - Anti-dependence
 - Output-dependence
- Control dependence

Dependence(cont')

- Data dependence
 - inst. i precedes inst. j
 - RAW hazard happen when inst. j use the result of inst. i
 - ~~Read After Write~~
 - Read Before Write
 - **RO** = M1 + R1;
R2 = **RO** + R3

Dependence(cont')

- Name dependency
 - There is **no data flow** between instructions
 - inst. i precedes inst. j

Dependence(cont')

- **Anti dependence**

- inst. i precedes inst. j
- inst. i and j use same register R0 at once.
inst. j **write** a value at R0 **before**
inst. i **read** it.
- Write After Read hazard happens
- **Read Before Write**

Dependence(cont')

- **Output dependence**
 - inst. i precedes inst. j
 - inst. i and j **use same register** R0 at once.
inst. j **write** a value at R0 before
inst. i write it
 - Write After Write hazard happens
 - **Write Before Write**
 - **Register Renaming** can be used for solving
this problem

Dependence(cont')

- **Control Dependence**

- all instruction dependent on the branch should depend on that branch.

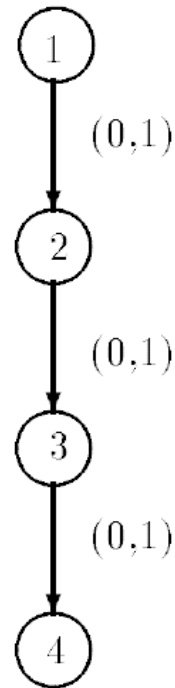
- ```
if(x > 0) {
 instructions j
} else {
 instructions i
}
```



# Data Dependence Graph

---

- The graph to express the dependences



# Data Dependence Graph

---

- Dependence Arcs
  - Loop independent arc dependences within same iteration.
  - Loop carried arc dependences between other iterations

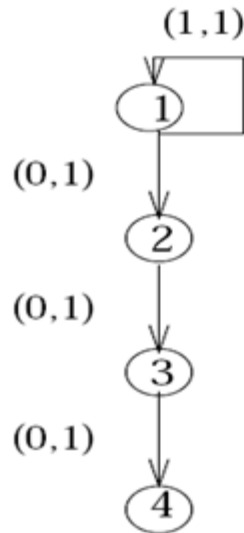
# Data Dependence Graph

---

- Loop independent
  - OP1 and OP2 are **loop independent**, if OP1 in Iter0 is not dependent OP2 in Iter1...N
  - Doall Loop

# Data Dependence Graph

- (diff, min)



Iter1

$$\$a[3] = a[2] + 1\$$$

$$\$b[2] = a[3] / 2\$$$

$$\$c[2] = b[2] + 3\$$$

$$\$d[2] = c[2]\$$$

Iter2

$$\$a[4] = a[3] + 1\$$$

$$\$b[3] = a[4] / 2\$$$

$$\$c[3] = b[3] + 3\$$$

$$\$d[3] = c[3]\$$$

# Software pipelining

- a techniques that reforms the loop so that a faster execution rate is realized
- $\{ABC\}$  is equivalent to  $A\{BC\}^{n-1}BC$

