Register Renaming

ECE 404
High-Performance Based Microprocessor System

Raj Parihar
parihar@ece.rochester.edu
Write After Write (WAW)

Name Dependence

- No data transmitted between two instructions
- Poor choice of registers or back-pressure
- Write After Write (WAW)

Instruction 1

- ADD R1, R2, R3
- SUB R1, R4, R5

Instruction 2

- ADD R1, R2, R3
- SUB R6, R4, R5
Write After Write (WAW)

- Name Dependence
  - No data transmitted between two instructions
  - Poor choice of registers or back-pressure
  - Write After Write (WAW)

```
ADD R1, R2, R3
SUB R1, R4, R5
```

Renaming:

```
ADD R1, R2, R3  
SUB R6, R4, R5
```
Write After Read (WAR)

Name Dependence

- No data transmitted between two instructions
- Poor choice of registers or back-pressure
- Write After Read (WAR)

```
ADD R2, R1, R3
SUB R1, R4, R5
```

Renaming

```
ADD R2, R1, R3
SUB R6, R4, R5
```
Read After Write (RAW)

- True Dependence
  - Data is transmitted between two instructions
  - Programmer is forced to re-use the register
  - Read After Write (RAW)

```
ADD R1, R2, R3
SUB R5, R1, R4

ADD R1, R2, R3
SUB R5, R6, R4
```

Renaming Won’t Help

Correctness suffers!
Conclusion

- Register Renaming should be done only in NAME dependence
- It can’t help in case of TRUE dependence
Register Renaming

- Dynamically eliminates name dependences in set of instructions registers
  - By avoiding WAR and WAW hazard
  - Can be done statically (= by compiler)
  - Or dynamically (= by hardware)