

Introduction

L'obiettivo è quello di ottenere un **CPI** minore di 1, ovvero eseguire più istruzioni con un solo ciclo di clock.

Due tipi di processore possono fare ciò:

- **superscalari**
- **VLIW**: processori vettoriali (istruzioni in parallelo)
in ambi i casi ovviamente le architetture contengono più **unità funzionali**.

Multiple Issue statico - MIPS superscalare

Torniamo al classico processore MIPS (senza architettura di Tomasulo).

Vengono *issued* più istruzioni nello stesso *clock cycle*.

Ciò si può fare se una delle istruzioni è un ALU e l'altra è una FP (load e store appartengono sempre ad ALU).

Due istruzioni (dunque 64 bit) vengono fetchate e decodificate ad ogni ciclo di clock.

Questo blocco di istruzioni da 64 bit viene allineato appunto a 64 bit e viene detto **issue packet**.

Le istruzioni di salto vengono eseguite **sempre** da sole

FP Units

Per ottenere un beneficio, le **unità FP** devono essere **pipelined** oppure **multiple** e indipendenti, per evitare che facciano da *bottleneck*

FP Register contention

Se una istruzione è una **FP load/store**, la porta del register file viene usata da tutte e due le istruzioni quindi:

- si aggiunge un'altra porta
- l'esecuzione viene fatta in ordine

RAW Hazards

Se la prima istruzione è una **FP load/store** e l'altra legge il suo risultato, potrebbero capitare dei **RAW hazards**.

In questo caso la seconda istruzione viene *deleyed* di un *clock cycle*.

Data e branch delay

Il delay slot(e branch delay slot) viene aumentato a 3 istruzioni per supportare il multi-issue
3 perchè 1+2 dove 1 è il delay che ci mette la load a rendere disponibile il dato

Conclusioni

I processori con multiple-issue e scheduling statico vengono utilizzati per il mercato degli embedded

Multiple Issue dynamic scheduling

Si ottiene adottando uno schema simile a quello di **Tomasulo**.

Per avere una implementazione più facile, le istruzioni non vengono **mai issued** alle reservation stations *out-of-order*

CDB criticality

Nello stesso ciclo di clock più istruzioni possono scrivere sul **CDB**. Bisogna dunque duplicare il **CDB** a costo di un overhead.