Politechnika Wrocławska

Architektura Systemów Komputerowych

Wykład 11

Dr inż. Radosław Michalski
Katedra Inteligencji Obliczeniowej, Wydział Informatyki i Zarządzania
Politechnika Wrocławska

Wersja 1.1, wiosna 2018

Źródła i licencja

Najbardziej aktualna wersja tego wykładu znajduje się tu:
https://github.com/rmhere/lecture-comp-arch-org

Zawartość tego wykładu

Liczby zmiennoprzecinkowe

Zaawansowane architektury

Liczby zmiennoprzecinkowe

Flagi warunków

- ustawianie flag w zależności od wyniku porównań
  - `c.eq.d $f0 $f2`
  - Jeśli zawartość rejestrów $f0 i $f2 są identyczna, ustaw flagę 0 na `true`.
  - `c.eq, c.lt, c.le`
- Branching w zależności od statusu flag
  - `bclt label`
  - Skocz do `label` gdy flaga `0 == true`
  - `bclt, bclf`

Liczby zmiennoprzecinkowe

Konwersja, przenoszenie

Konwersja:

- `cvt.x.w $fd $fs` - l. całkowita do zmiennoprzecinkowej
- `cvt.w.x $fd $fs` - l. zmiennoprzecinkowa do całkowitej
- `cvt.s.d $fd $fs` - l. podwójnej precyzji do pojedynczej
- `cvt.d.s $fd $fs` - l. pojedynczej precyzji do podwójnej

Przenoszenie:

- `mfc0, mfc1` - przenoszenie z koprocesora 0 lub 1
- `mtc0, mtc1` - przenoszenie do koprocesora 0 lub 1

Zaawansowane architektury

W jaki sposób zwiększać wydajność systemu?

- pomysły?

Zaawansowane architektury

Optymalizacja

- szybkość
- zużycie energii
- cena
- niezawodność

Zaawansowane architektury

Minimalizacja tranzystorów

- tranzystory co 2-3 lata zmniejszają rozmiary o ok. 30%
- dzięki temu:
  - działają szybciej
  - można ich umieścić więcej w układzie
- jednak zwiększa to zużycie energii
- inne problemy (np. ścieżki - artykuł)

Zaawansowane architektury

Głębokie potokowanie

- zamiast typowych pięciu etapów potokowania - więcej
- przykładowo, procesor Pentium 4 miał od 20 do 31 etapów
- hazard potokowania będzie częściej występował
- znowu problem zużycia energii
- powrót do mniejszej liczby etapów

Zaawansowane architektury

Przewidywanie rozgałęzień

- problem rozgałęzień w potokowaniu
- przewidywanie rozgałęzień - jednobitowe i dwubitowe

Zaawansowane architektury

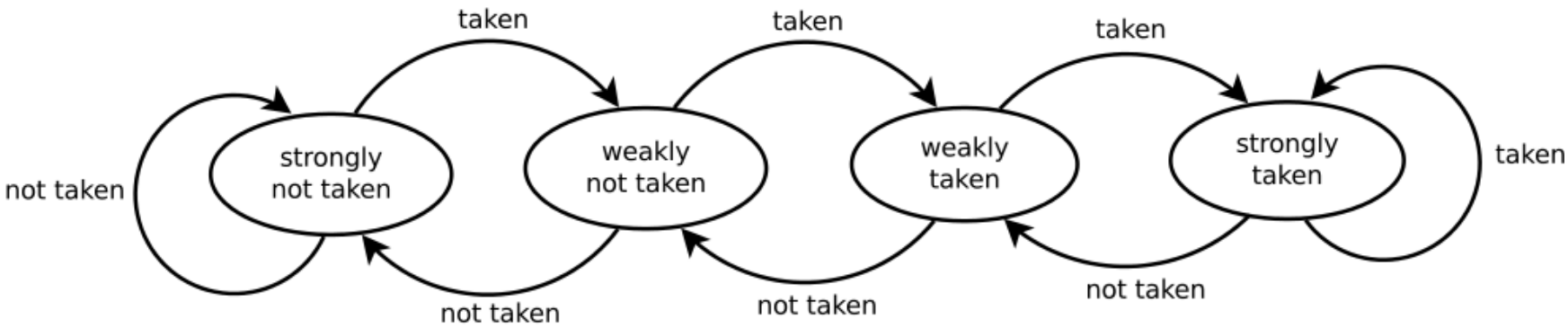
Przewidywanie rozgałęzień - przykład

```
add $s1, $0, $0
addi $s0, $0, 0
addi $t0, $0, 10

for:
    beq $s0, $t0, done
    add $s1, $s1, $s0
    addi $s0, $s0, 1
    j for
done:
```

Zaawansowane architektury

Przewidywanie rozgałęzień - jednobitowe a dwubitowe



Enormator, *CC-BY-SA-3.0*

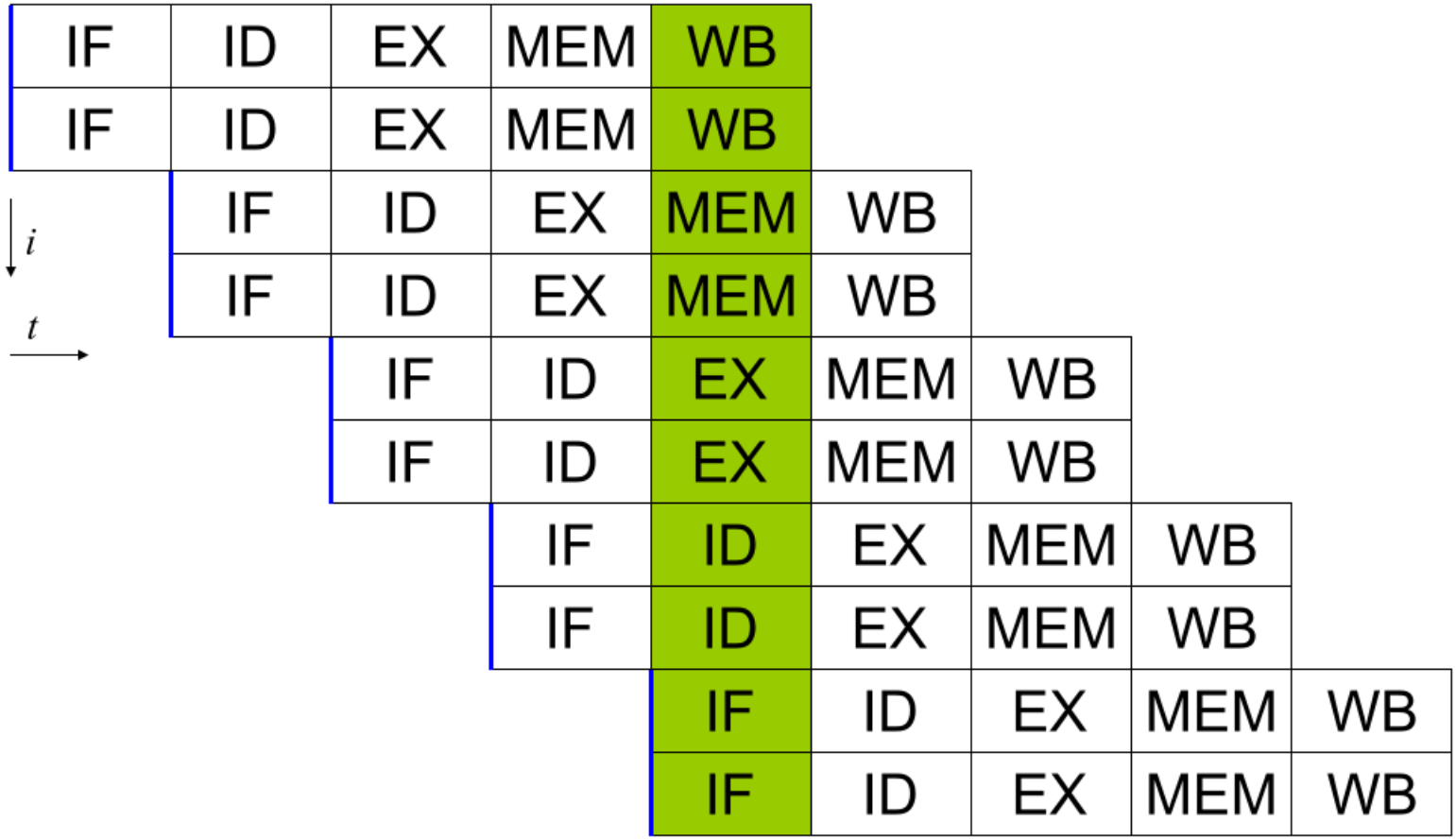
Zaawansowane architektury

Superskalar

- zwielokrotnienie szyny danych
- wiele jednostek wykonawczych
- zależności

Zaawansowane architektury

Superskalar - potoki



Zaawansowane architektury

Tranzystor

Wideo

AT&T Tech Channel - The Transistor: a 1953 documentary, anticipating its coming impact on technology

Zaawansowane architektury

MegaProcessor

Wideo

Computerphile - MegaProcessor

Slajd końcowy

Źródła i polecane materiały

- Katherine Boursac, „Graphene could Buttress Next-gen Computing Chip Wiring”, IEEE Spectrum (artykuł)
- David Harris and Sarah L. Harris, „Digital Design and Computer Architecture”, Morgan Kaufmann (książka)

Slajd końcowy

Pytania? Komentarze?

Jeśli masz pomysł jak poprawić lub wzbogacić te wykłady,
proszę zgłoś to jako issue w tym repozytorium:

https://github.com/rmhere/lecture-comp-arch-org