

02

CPU

Leistung

- Viele Befehle => Befehle zur Berechnung eines Problems sinkt
 - Anzahl Befehle hängt vom Compiler und Befehlssatz der CPU ab
- Taktzyklus(zeit) (Angabe als Frequenz): e.g. 4GHz = 0.25ns
 - Kleine Taktzykluszeit => viele Befehle können pro Zeiteinheit ausgeführt werden
 - Abh. von CPU-Implementierung
- CPI := clock cycles per instruction (Taktzyklen pro Befehl, d.h. \varnothing Anz. Taktzyklen, welche ein Befehl zur Ausführung benötigt)
 - Abh. von CPU-Implementierung
- Komponenten = {Befehlzähler/Befehlsregister, Steuerwerk, Register, ALU=Rechenwerk} // inkl. dazugehörige Steuer-, Daten-, Adressleitungen, Datenlogik, Speicher (entspricht d. Von-Neumann-Rechner-Architektur)
 - Für die Bearbeitung von Befehlen wird auf den Speicher zurückgegriffen
 - Befehle führen arithmetisch-logische Funktionen aus (mittels ALU ~ Arithmetisch Logical Unit)
- Zykluszeit (Faktoren)
 - Architektur, Hardware, Befehlssatz so designen, dass alle Befehle gleich lange brauchen (zur Ausführung)
 - Befehle in Gruppen unterteilen, die in etwa gleich lang brauchen (für Ausführung)
 - Pipelining: Mehrere Befehle überlappend/zeitgleich ausführen
 1. Befehl wird geladen
 2. Befehl wird decodiert (Steuerwerk)
 - Operanden werden bereitgestellt
 - Rechenoperation wird durchgeführt (ALU) + Ergebnis wird geschrieben
- Pipelining
 - structural hazard := Gleichzeitiger Zugriff auf Ressourcen durch aufeinanderfolgende Befehle
 - data hazard := Befehl B greift auf Daten des Befehls A zu, wobei A noch nicht abgeschlossen ist

- control hazard :=
- Branch prediction: Sprungbefehle können zu >90% vorhergesagt werden, Ursache: meist eingesetzt bei Schleifen. e.g. bei n Durchgängen wird n-1 mal richtig geraten, beim Verlassen des Loops falsch geraten. Daher Fehlerquote <10%!

03

Einführung

- Wörter ~ Befehle, Sprache ~ Befehlssatz (instruction set)
- Intelligenz im Rechner steckt im Programm (Rechner wird von Befehlen gesteuert)
- Video: Security Phorensiker, der Viren mit public domain sw programmiert hat, welche via bluetooth o.ä. eingeschläust werden kann, auf HW-Ebene. Theoretisch möglich: HW manipuliert (China), SW manipuliert bzw. OS (NSA)
- Kryptographie/Internetsicherheit (als Vertiefung)
- Begriff := Rechner berechnet arithmetisch-logische Funktionen (e.g. arithmetisch: Addition, logisch: OR)
 - e.g. OP-Code, Operand-1, Operand-2, Operand-3, (Option)
 - e.g. ADD, a, b, s, - // s=a+b, keine Option (-)
 - e.g. OR, u, v, w, - // w = u OR v, keine Option (-)
 - e.g. ADD, R-1, 100, R-2 // Addition mit Register (R-1, R-2) und Wert 100 (Schreibe (Inhalt R-1)+100 nach R-2)
- Register: sehr schnell, teuer, hardwaretechnisch gross (daher nicht beliebig viele einsetzbar). Daher werden arithmetisch-logische Funktionen oft mittels Operanden-Register berechnet.

Sprungbefehle

- Theoretisch 1 Sprungbefehl ausreichend, um alle abzudecken. Dennoch werden verschiedene implementiert, da somit der Umfang des Programmcodes erheblich reduziert werden kann (“Bequemlichkeit”)
- e.g. Bnull, R-1 := Branch if null
- e.g. IF Condition: ... Test 2x0 // test if condition true Jump // jump to line xy