

### 1.1

Embedded computer systemen zijn zo ontworpen dat ze maar een taak of een serie van taken kunnen doen. Omdat de processor maar een taak hoeft uit te voeren; en hij maar zo snel hoeft te zijn als voor die taak nodig is. Is het belangrijkste doel bij Embedded systems om de kosten en de stroom zo laag mogelijk te houden.

### 1.2

Om computerarchitectuur voor programmeurs begrijpbaarder te maken worden bepaalde details in de architectuur weggelaten. Op deze manier wordt de productiviteit verhoogd.

Om meer preformance te krijgen is het belangrijk om dingen die veel herhaald worden te optimaliseren; als iets maar een enkele keer voorkomt heeft dit praktisch geen nut.

Om meer preformance te krijgen kunnen er ook meerdere taken naast elkaar worden uitgevoerd dit wordt parallelism genoemd.

Een andere manier om meer preformance te krijgen is pipelining, een taak uit laten voeren door meerdere is efficiënter dan verschillende individuen een stukje van de taak uit te laten voeren.

Nog een manier om meer preformance te krijgen is door prediction. Het is sneller om ongeveer het antwoord te raden en te beginnen met werken dan wachten tot je het antwoord precies weet.

### 1.3

Hardware kan in principe alleen maar kleine en eenvoudige taken uitvoeren. System software vertaalt moeilijke processen naar simpele (binaire) instructies voor de hardware.

Het vertalen gebeurt aan de hand van een aantal soorten system software. Allereerst wordt de high-level programming language vertaald naar een assembly language, dit gebeurt door een compiler.

Vervolgens wordt deze assembly language omgezet naar binaire code door een assembler.

Het operating system verwerkt de in en output en zorgt ervoor dat de compiler zijn werk doet wanneer dit moet gebeuren.

### 1.4

De functies van de hardware in elke computer zijn hetzelfde; het invoeren van data, het uitvoeren van data, het verwerken van data en het opslaan van data.

In een computer vinden we een moederbord dat we grofweg in drie delen kunnen onderverdelen; het onderdeel dat het moederbord met de I/O apparaten verbindt, de processor en het geheugen.

De processor verwerkt de data die binnenkomt; het datapath voert de taken uit en het control stuurt het datapath aan.

Het geheugen slaat de data op. Er zijn twee soorten geheugen volatile (primary) geheugen, dat data verliest als het geen stroom krijgt en nonvolatile (secondary) geheugen dat zijn data kan behouden.

De instruction set architecture is de structuur van de hardware dat voor de programmeur van belang is om een goed programma te schrijven.

### 1.5

Chips worden gemaakt van siliconen. Door een chemische behandeling krijgen siliconen plakken de eigenschappen van een conductor insulator of een combinatie van beide. Chips worden op grote schaal gemaakt, maar een microscopisch kleine fout kan een groot aantal chips laten falen.

### 1.6

Performance van een computer wordt gemeten in tijd. Je kan kijken naar de execution time, de tijd die een computer erover doet om een taak te voltooien. Hoe lager de executietijd hoe hoger de performance.

CPU performance wordt gemeten aan de hand van de tijd die de CPU bezig is om een taak te voltooien. De CPU performance is voor programmeurs het meest interessant.

Om de performance goed te metten moet er naar verschillende zaken gekeken worden. De instruction count (de hoeveelheid instructies die moeten worden uitgevoerd) het CPI (Hoeveelheid

cycles die er nodig zijn om een instructie uit te voeren) en de clock rate (De hoeveelheid cycles die er per seconde worden uitgevoerd).

Verskillende aspecten hebben invloed op verschillende onderdelen van de performance. Zo hebben zowel het Algoritme, de programmeertaal en de compiler invloed op de Instruction count en het CPI. De architectuur heeft invloed op zowel de Instruction count, de clock rate en het CPI.

#### 1.7

De clock rate van processoren is enorm toegenomen de afgelopen tientallen jaren. Daarmee nam ook de hoeveelheid energie die hier voor nodig was toe. Omdat een te grote hoeveelheid energie niet goed gekoeld kan worden zijn de makers van chips tegen een zogenaamde power wall aangelopen. Chips kunnen niet sneller worden zolang de hoeveel energie die ze nodig hebben niet afneemt.

#### 1.8

De performance van microprocessoren wordt nu verbeterd door gebruik te maken van meerdere processoren tegelijkertijd te gebruiken; parallelism. Hierdoor moeten programmeurs hun code continu aanpassen om de beste performance te krijgen uit de multicore microprocessoren.

Om extra performance te krijgen door parallelism is het belangrijk dat de code goed geschreven is en de verschillende processoren ongeveer evenveel werk te doen hebben.

#### 1.9

Benchmarks kunnen gebruikt worden om performance van verschillende machines met elkaar te vergelijken.

#### 1.10

Amdah's Law beschrijft dat een verbetering in performance van een bepaalde feature, de verbetering gelimiteerd is tot hoeveel die feature gebruikt wordt. Make the common case fast.

MIPS is een andere manier om performance te meten. Deze kijkt naar hoeveel miljoen instructies per seconde worden voltooid. Een hogere MIPS betekent echter niet altijd een snellere computer.