

Datorteknik

Tomas Nordström

Föreläsning 13

Kurssammanfattning

Kurssammanfattning



- Lecture1 - Intro och datarepresentation
- Lecture2 - Datorarkitektur, Instruktionscykeln
- Lecture3 - Assembler och ARM
- Lecture4 - Assemblerprogrammering
- Lecture5 - Subrutiner och Stack
- Lecture6 - IO (Periferienheter)
- Lecture7 - Avbrott
- Lecture8 - Timers
- Lecture9 - Programmering och systemutveckling
- Lecture10 - Accelerationsmekanismer; Pipelining
- Lecture11 - Minneshierarki
- Lecture12 - I/O (AD- och DA-omvandling; SerIELkomm.)

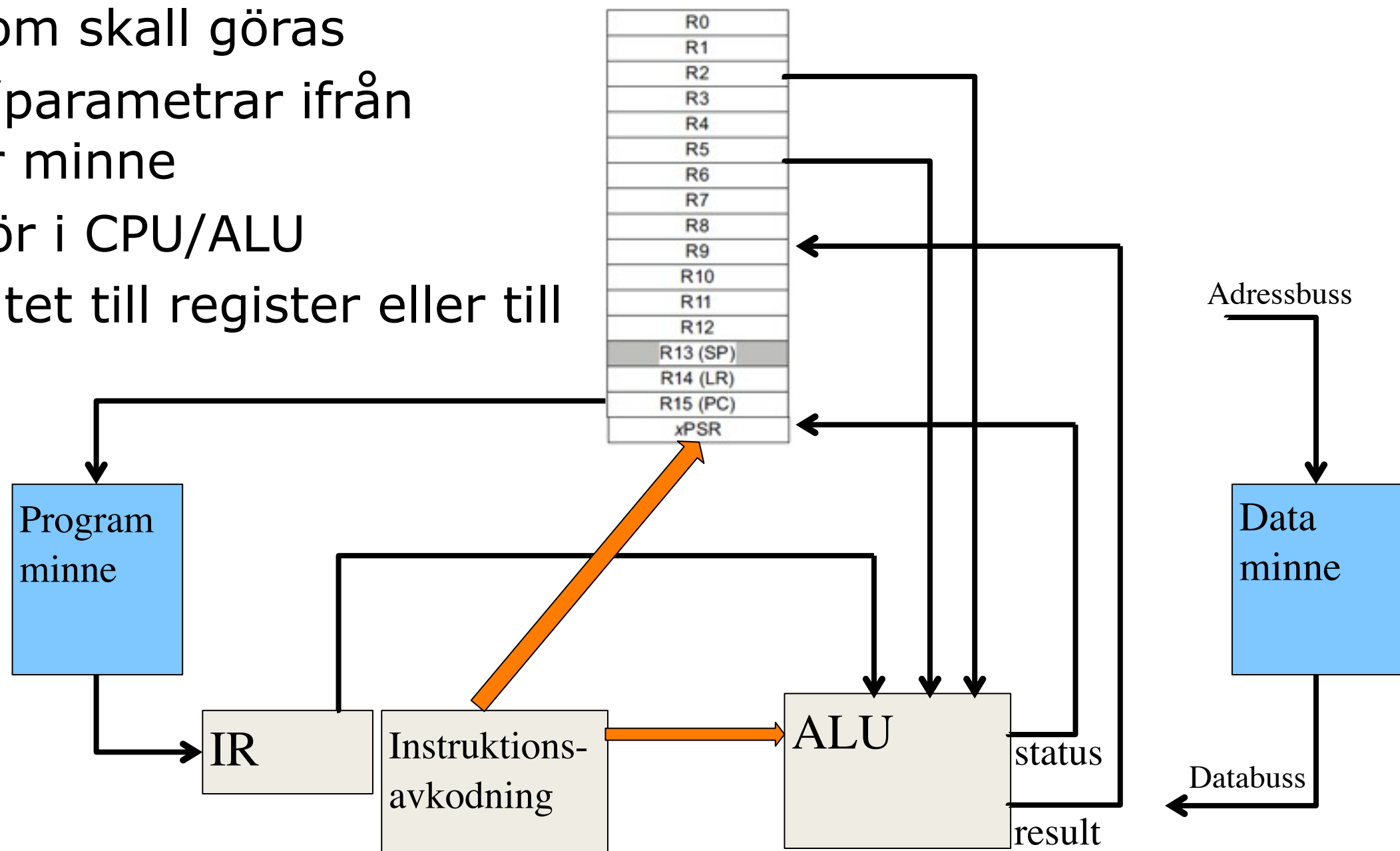
Förstå hur en dator är uppbyggd



- Datorn byggs upp av de komponenter ni lärt er i digitalteknik <register (D-vippa), adderare, aritmetiska och logiska enheter, multiplexer, avkodare och kodare, minnen>
- Programmet, datorns instruktioner ligger i minnet - von Neumann arkitektur
- Hur dessa ettor och nollor i minnet kan representera olika saker som tex tal (med och utan tecken, decimaltal och flyttal), (ascii) tecken, instruktioner
- Instruktionscykeln
 - ▶ Fetch, Decode, Execute

Arbetsgång Styrenhet

1. Hämta instruktion från minnet
2. Avgör vad som skall göras
3. Hämta data/parametrar ifrån register eller minne
4. Beräkna/utför i CPU/ALU
5. Skriv resultatet till register eller till minnet



Mer Arkitektur

- Pipelining (och Superpipelining)
- Minneshierarki

I/O för att interagera med omvärlden

- Periferienheter (tex PIO, SYSTICK, TIMER/COUNTER)
- Parallella/Seriella gränssnitt/kommunikation
- AD och DA omvandling
- Men även förstå hur ett minne/minneskapsel accessas för att hämta data eller instruktioner
- Pulling/Avbrott

Programutveckling

- Förstå problemet, beskriv kraven
 - Förstår man vilka I/O signaler som finns och vad de ska göra så har man kommit långt!
- Modularisering (subrutiner, avbrottshanterare)
 - Hitta delar som man skulle vilja abstrahera bort och göra lite senare
 - Hitta kodavsnitt som upprepas -> lämpligt som subrutin?
 - Hitta rutiner som du kan tänka dig dyker upp i andra program/problem
- Översätta till en algoritm (flowchart, pseudokod)
- Skriva koden
- Dokumentera
- Testa (helst varje modul/subrutin för sig)

Exempel

- Vi vill utveckla ett system som kan mäta tiden ifrån det att en flaska lyfts ifrån en platta till den sätts ner igen.
- Funktionsspecifikation
 - Tiden skall mätas med tusendelar men visas bara med hundradelar.
 - Då tidsmätning pågår ska den kontinuerligt uppdateras på displayen och en lysdiod ska lysa
 - Då tidsmätningen upphör ska tiden fortfarande visas tills resetknappen trycks ner
 - Vid reset ska alltid tidsräkningen stoppas och klockan nollställas (00:00)
- I/O specifikation
 - Det finns en plattsensor kopplad till PIOA pinne 7 (1= något på plattan, annars 0)
 - Det finns en knapp kopplad till PIOA pinne 6 att användas som reset av vår tidsmätning (0= knapp nedtryckt, 1= ej nedtryckt)
 - Det finns en lysdiod kopplad till PIOA pinne 0 (0= lysdioden lyser, 1= lyser inte)
 - Vi får även anta att det finns en 7-segmentsdisplay inkopplad med fyra siffror som kan visa tiden. Och att någon skrivit en subrutin som tar en inparameter i R0 som representerar 4 ASCII siffror packade i 32-bitar och visar dem på displayen.