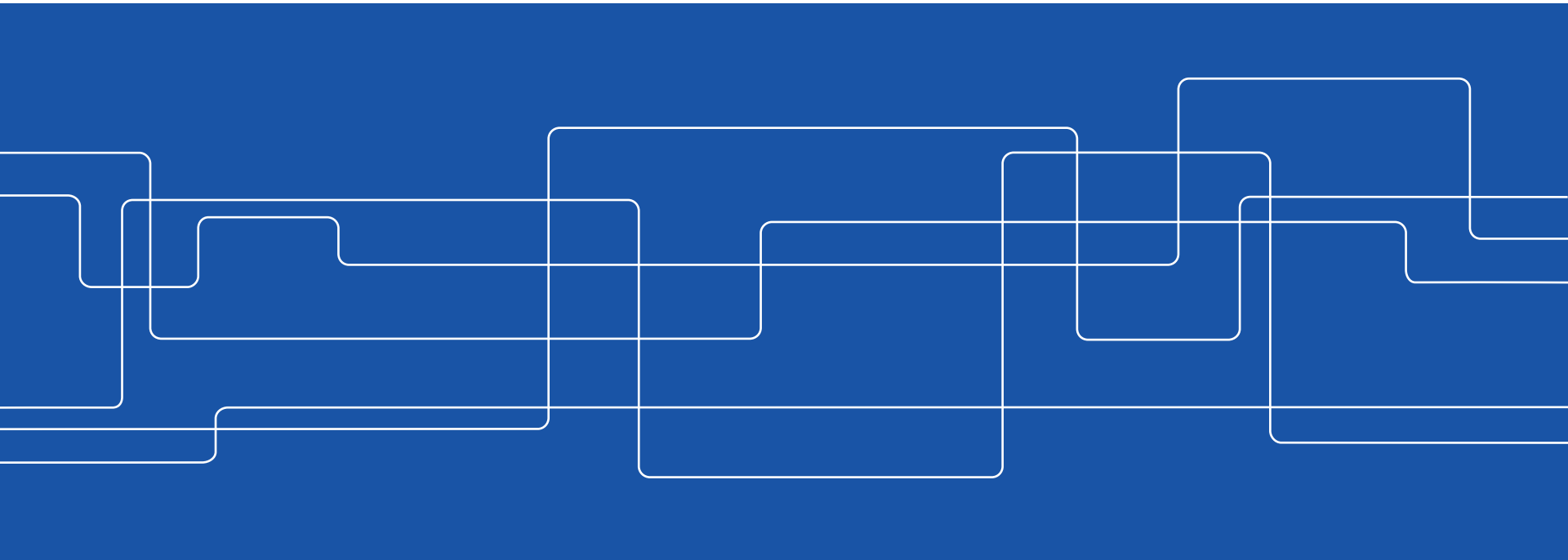
HF0010

Introduktionskurs i datateknik 1,5 hp



Välkommna

- till KTH, Haninge, Datateknik, kursen och till första steget mot att bli programmerare!

Er lärare och kursansvarig: Nicklas Brandefelt, bfelt@kth.se
Assistent: Johan Pettersson

All information finns på KTH- social. Enklast är att söka på HF0010 (https://www.kth.se/social/course/HF0010/page/tidaa-4/)

Kursen består av 3 föreläsningar och 4 laborationstillfällen.
För att bli godkänd på kursen krävs godkänt på 3 laborationer. Godkänd blir man genom att närvara vid laborationstillfällena och redovisa de tre laborationerna.

Dock är kursens syfte är att förbereda er för era studier och främst grundkursen i programmering så ta tillfället i akt och lär er så mycket som möjligt.

Föreläsningar och laborationer

F1 Datalogi och datorer

F2 Operativsystem och programmering

Lab1 Programmering med grafiska byggblock

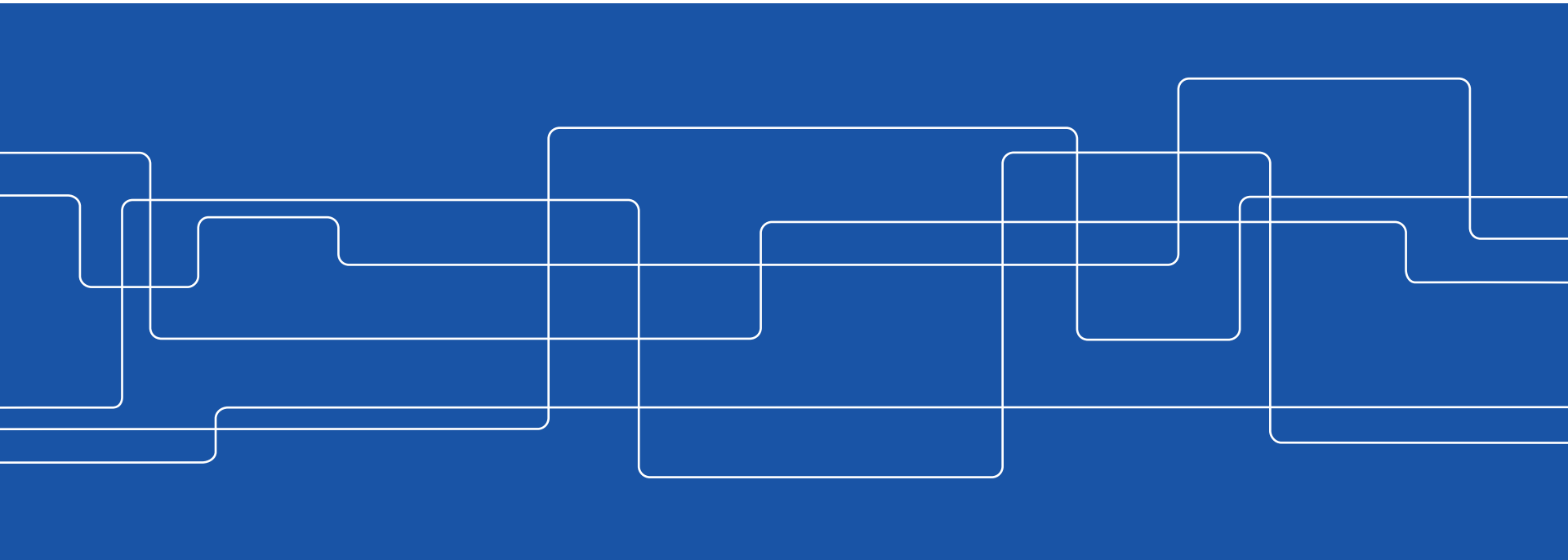
F3 Programmering, C och programmeringsmiljö

Lab2 Programmering i Javascript – Ett första spel

Lab3 Installera en programmeringsmiljö på din dator och skriv ditt första C-program.

Föreläsning 1

Datalogi och datorer



Datalogi (datavetenskap)

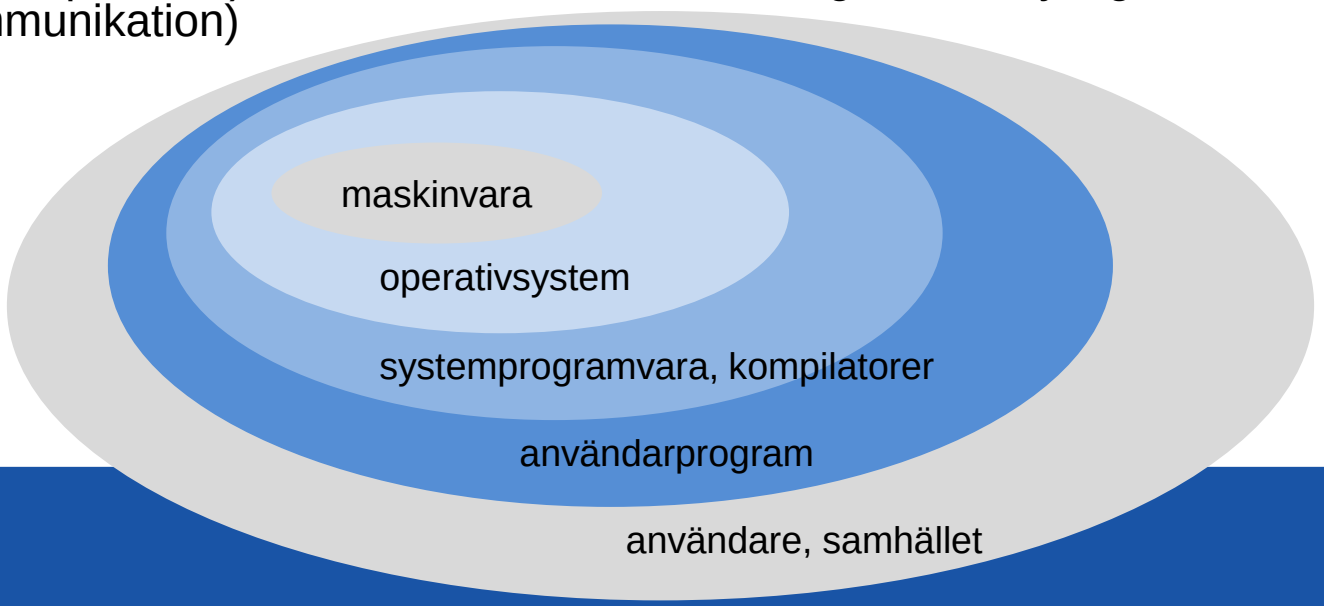
- saknas vedertagen definition

Ett synsätt:

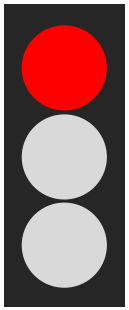
> Inom datalogin studeras hur data lagras, förmedlas, används och bearbetas ur alla aspekter

- den vetenskap som ligger till grund för stora delar av er utbildning

Vissa delar av datalogin är mogna och står på stark teoretisk grund (ex algoritmer, kompilatorer) medans andra är mindre mogna och otydligare (ex datakommunikation)



Information - data



Trafiksignalen är ett datamedium

Den innehåller data (röd lampa lyser)

Vi tolkar datat som att det är en bra ide att stanna – information

Data är potentiell information – men det krävs en tolkning för att det ska bli information

För en tolkning krävs konventioner - tolkningsföreskrifter

Dessa kan vara situationsberoende

Representation av data

Samma fakta kan representeras på olika sätt

- analog/digital klocka

Klassifikation av representation av data:

analog data – diskret data (digital data)

Positionssystemet

En symbols värde beror på dess position

Decimala talsystemet – basen 10 – 0,1,2,3,4,5,6,7,8,9

$317{,}4$ tolkas som $3\cdot10^{2} + 1\cdot10^{1} + 7\cdot10^{0} + 4\cdot10^{-1}$

Binära talsystemet – basen 2 – 0,1

1011 tolkas som $1\cdot2^{3} + 0\cdot2^{2} + 1\cdot2^{1} + 1\cdot2^{0}$

$= 1\cdot8 + 0\cdot4 + 1\cdot2 + 1\cdot1 = 11$

Skrivs: $(1011)_2 = (11)_{10}$

$(173)_{10} = (?)_2$

173 / 2 = 86 rest 1
ger att sista siffran är 1
jmf 173 / 10 = 17 rest 3 ger att sista siffran är 3

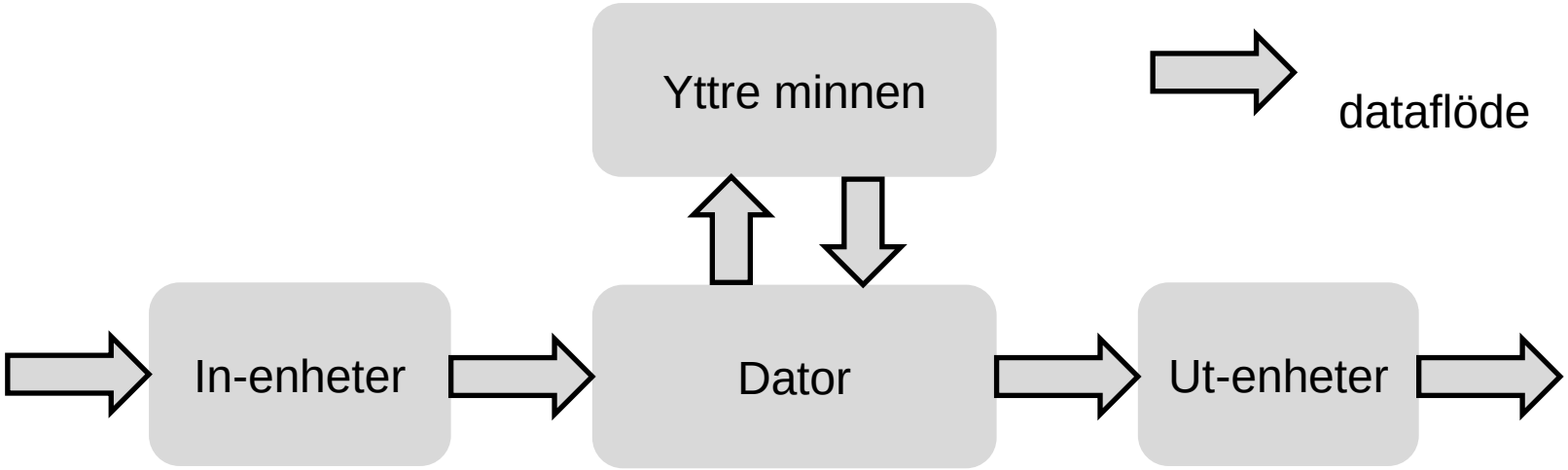
173 / 2 = 86 rest 1
 86 / 2 = 43 rest 0
 43 / 2 = 21 rest 1
 21 / 2 = 10 rest 1
 10 / 2 = 5 rest 0
 5 / 2 = 2 rest 1
 2 / 2 = 1 rest 0
 1 / 2 = 0 rest 1

läs av nerifrån och upp

$(173)_{10} = (10101101)_2$

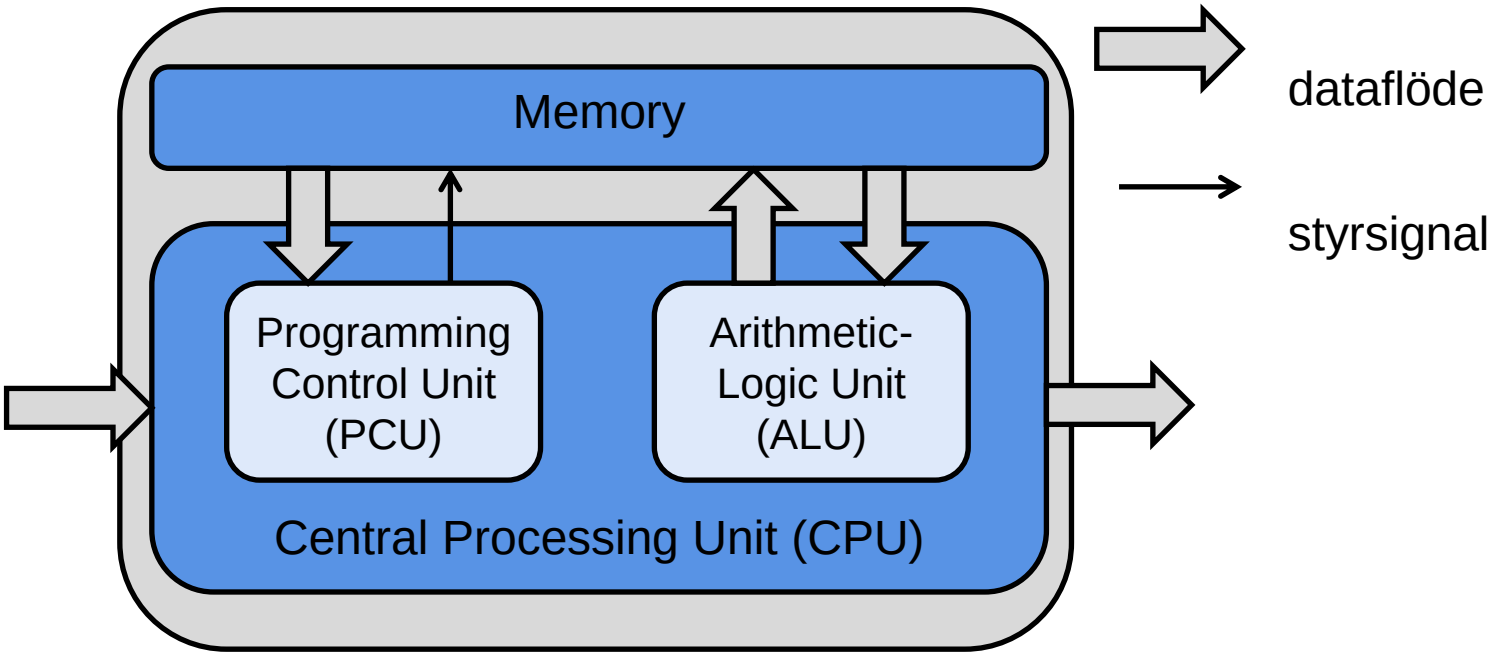
Datorn

En dator är en maskin som kan bearbeta data automatiskt. Datorn kan bestå av mekaniska, elektroniska, optiska eller andra komponenter och styrs med instruktioner.



Datorarkitektur

Von-Neumann-arkitektur: datorns instruktioner i samma minne (address-rymd) som datorn använder för att lagra indata och mellanresultat.



Prestanda

Prestanda hos processorn bestäms av klockfrekvensen, tillgängliga instruktioner och antal klockcykler som en instruktion tar att exekvera.

Klockfrekvensen är antalet enkla instruktioner en processor kan göra per sekund och ligger idag på storleksordningen 2 GHz. Dock tar många instruktioner fler än en cykel att exekvera.

Prestanda hos datorn påverkas dock av flera andra faktorer såsom läshastighet hos minnet och busshastighet för data och styrsignaler.

Minne

Under datorns utveckling har ofta hastigheten hos minnet varit en flaskhals för datorns prestanda. **Primärminnet** eller arbetsminnet (DRAM-minne) där datorn lagrar program och data har varit dyrt då man försökt göra det så snabbt som möjligt. Detta har då kompletterats med ett **sekundärminne** (hårddisk, flash) där delar av programmet och data som inte just nu behövs har kunnat swappas ut för att hämtas in när det behövs. Sekundärminnet är långsammare men mycket billigare och kan därför var mycket större.

Cacheminne

Då processorerna blev snabbare och snabbare hängde inte hastigheten hos primärminnet med. Lösningen blev att skapa ytterligare ett lager. Ett litet men mycket snabbt minne, cacheminnet, placerades mellan processor och primärminne. Viktigt för prestandan är då att datorn i förväg lyckas ”gissa” vad den bör fylla cacheminnet med så att processorn inte får en cachemiss och måste stå och vänta tills rätt minnescell hunnit läsas från primärminnet. Dagens datorer har oftast flera nivåer cacheminnen.

För att skriva effektiva program är det viktigt att vi undviker cachemissar. Det gör vi tex genom att gå igenom matriser i ”rätt” ordning.

Minne kan delas upp i RWM (read and write) och **ROM** (read only). RWM kallas oftast **RAM** (random access memory) vilket egentligen betyder att man kan nå godtycklig del av minnet direkt och inte behöver läsa sekventiellt. ROM-minnet är beständigt även när strömmen sluts och används t.ex för att lagra instruktioner (BIOS) som startar upp datorn när den slås på. Idag finns beständiga minnen som kan skrivas till (PROM, EPROM, EEPROM, flash).

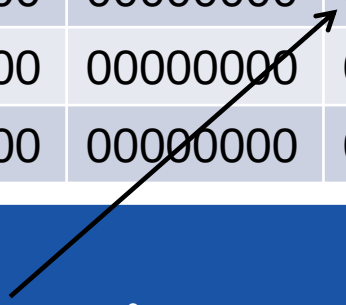
Primärminnet

Primärminet består av ett antal celler. Varje cell innehåller ett ord (word) och har en unik adress. Ett ord består av ett visst antal bitar (bits). En bit har antingen värdet 0 eller värdet 1. Ordlängden kan t.ex vara 8, 16, 24, 32, 64. Persondatorer har idag ordlängden 32 eller 64 bitar.

Med hjälp av adressen till en minnescell kan man komma åt innehållet i cellen (ordet, datat).

Ordet består då av en rad ettor och nollor. För att processorn ska kunna använda dessa data måste den veta hur de ska tolkas (som ett heltal, decimaltal, bokstav,…).

00110101	00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000	00000000
00000000	00000000	00001011	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000	00000000



Nås med adress 13 (binärt 1101)

Styrenheten (PCU)

Styrenheten hämtar en instruktion i taget från primärminnet och utför den. Den har till sin hjälp ett antal register.

Aritmetiska enheten (ALU)

Innehåller ett antal arbetsregister. Skall den tex beräkna a + b och tilldela detta till c kopieras först a’s värde till ett arbetsregister sedan adderas b’s värde till detta och sedan kopieras resultatet till c’s plats i minnet.

Dataöverföring

Dataöverföring mellan datorns delar sköts av ett bussystem som består av två bussar:

en adressbuss som överför vilka adresser som är aktuella från styrenheten till primärminnet. Bredden avgör hur mycket minne som kan adresseras.

en databuss (minnesbus) som står i förbindelse med samtliga datorns enheter och är gränssnitt mot yttre världen via in och ut-register. Bredden är ofta lika med ordlängden och om den är 32 bitar överförs dessa parallellt.

Vid läsning överförs det ord vars adress finns på adressbussen till databussen där det kan hämtas av den enhet som behöver ordet.

Vid skrivning överförs det ord som finns på databussen till den cell vars adress finns på adressbussen.

Harvard-arkitektur

Till skillnad från persondatorer har de flesta inbyggda system inte von-Neumann-arkitektur utan Harvard-arkitektur. Det innebär att de har ett separat minne (med separat adressrymd) för instruktioner med separat bus. Detta kan då vara ROM (read only memory). Bland annat möjliggör detta snabbare behandling då instruktioner kan läsas parallellt med data. Man kan också ha olika ordbredd på de två minnena.

Laboration 1

På en timme lär ni er programmera enkla program med grafiska byggblock i hour of code eller timmen med kod. Gå in på hemsidan:

https://studio.code.org kicka på: Timmen med…

När du är klar visar du lärare eller assistent diplomet så att vi kan bocka av dig.

På nästa svårighetsgrad använder vi samma block för att rita:

https://blockly-games.appspot.com/turtle?lang=en&level=1

Visa din kod när du löst uppgift 5. Du har nu klarat laboration 1.

Nu fortsätter du med resten av uppgifterna i mån av tid.