Designskiss - The Gentoo Saga

TSEA83 Grupp 33

Emil Segerbäck - emise935 - emise935@student.liu.se
Malcolm Vigren - malvi108 - malvi108@student.liu.se
Robin Sliwa - robsl733 - robsl733@student.liu.se

2016-04-05

Innehållsförteckning

1 Inledning **2**

2 Spelet **2**

3 Analys av problemet **2**

4 Blockscheman **3**

5 Milstolpe **5**

1 Inledning

Vi ska göra en dator som kör spelet The Gentoo Saga. Användaren spelar på ett tangentbord som kommunicerar med datorn via PS/2 och spelet visas på en VGA-skärm via dedikerad grafikhårdvara. Denna hårdvara består av en grafikmotor, tileminne och spriteminne. Datorn ska också spela musik och eventuellt ljudeffekter (om vi har tid) via dedikerad ljudhårdvara.

Processorn är av pipeline-typ, liknande den som användes i Pipeline-labben, vilket innebär separata program- och dataminnen.

2 Spelet

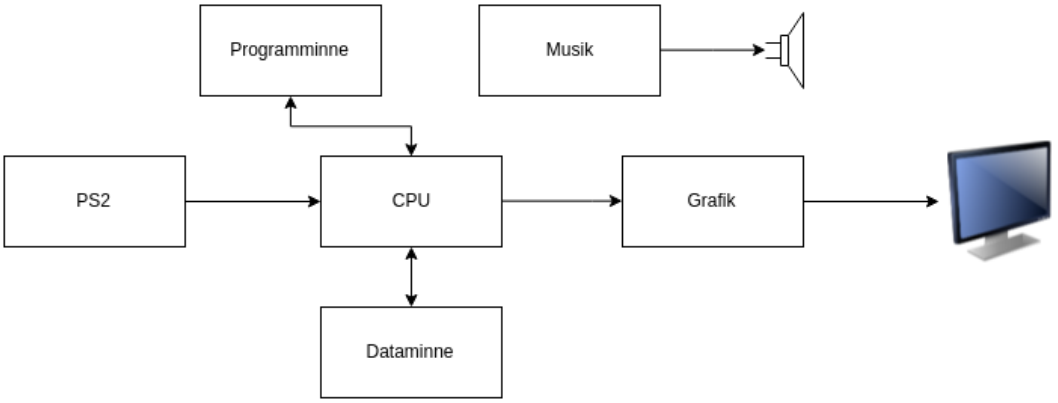
Spelet går ut på att spelaren ska styra en pingvin som ska försöka nå fram till en Gentoo-logga i slutet av banan, utan att springa på arga monster eller ramla av banan på vägen dit. Pingvinen styrs med hjälp av piltangenter där upp används för att hoppa, och höger och vänster är till för att gå till höger respektive vänster på skärmen. Banan som pingvinen går på har hål i marken som spelaren kan ramla igenom. Om man hoppar ovanpå ett monster försvinner det. Om pingvinen nuddar ett monster från sidan eller ramlar ut ur banan så tar spelet slut. Om man däremot lyckas ta sig till Gentoo-loggan så klarar man banan.

3 Analys av problemet

- **Inmatning**: Modulen som läser in data från tangentbordet skriver till ett speciellt register i CPU:n.
- **VGA (bildvisning)**: Vi ska ha en bild på 320x240px vilket resulterar i att en storpixel är 4 (2x2) småpixlar. Genom att göra detta kommer vi att ha mer tid att rita ut varje pixel på skärmen.
- **Musik**: Vi ska ha hårdvara som läser musik från speciellt musikminne. Den stegar genom noterna som ligger i minnet och spelar upp dem i högtalaren.
- **Minne**: Alla register är 32 bitar breda. Vi använder ett programminne, dataminne, tileminne, ett extra tileminne för bakgrund, spriteminne och musikminne. Tileminnet har 15x150 tiles. Vi har 32 olika tiles så tileminnet behöver 15x150x5=11250 bitar=1407 bytes. Det andra tileminnet behöver 15x75x4=4500 bitar=563 bytes. Spriteminne behöver bara innehålla två sprites på 16x16 pixlar. Musikminne behöver instruktioner som har 5 bitar för tonhöjd och 2 bitar för tonlängd, totalt 8x128=1024 bitar=128 bytes. Programminnet behöver 2000 instruktioner totalt 32x2048=8192 bytes. Till dataminnet tar vi 2048 bytes. FPGA-kortet har 32 blockram, 2kB styck, vilket är mycket mer än vår uppskattning. Därför tror vi att minnet kommer att räcka.

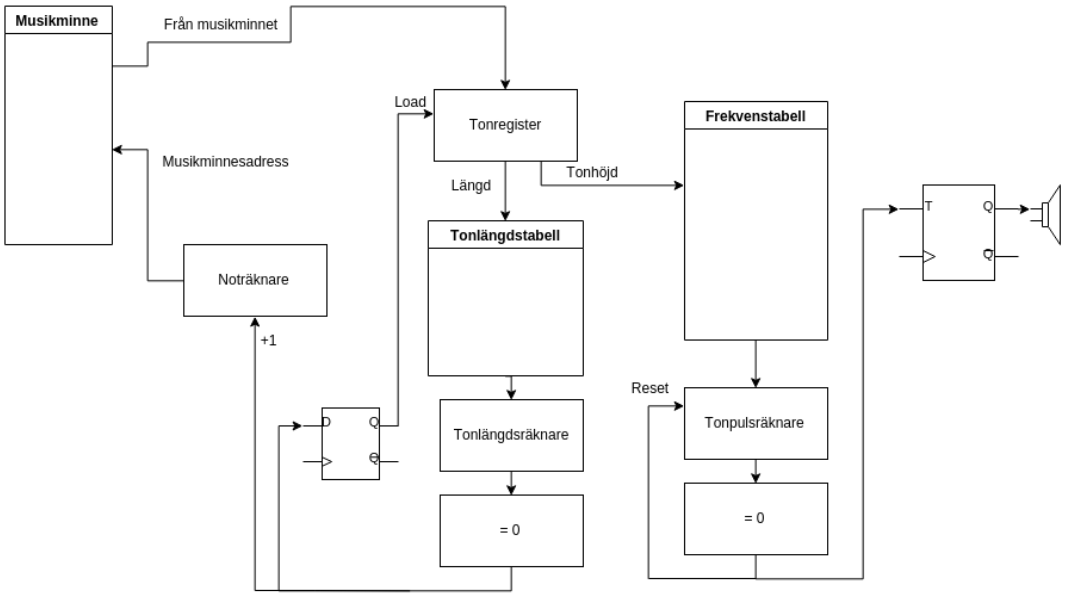
4 Blockscheman

Musikenheten kommer till en början inte att kommunicera med CPUn, kan ses i Figur 1, men som ett börkrav har vi ljudeffekter. Detta innebär att om vi senare hinner lägga till ljudeffekter så kommer musikenheten behöva kommunicera med CPUn.

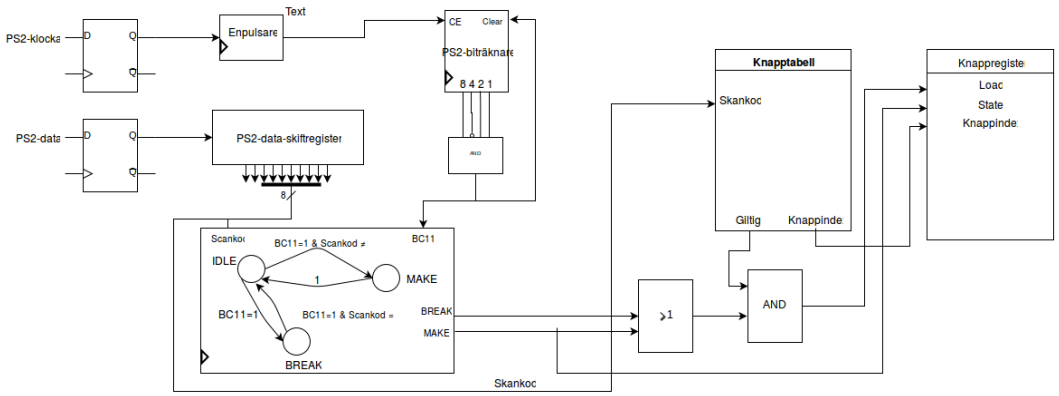


Figur 1: Översiktligt blockschema

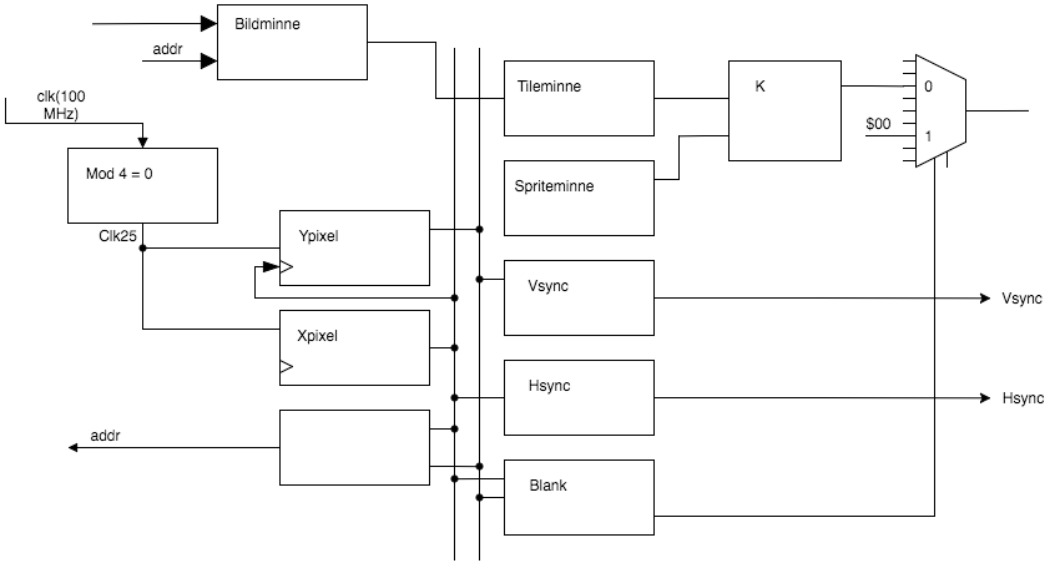
Musikmodulen loopar igenom ett speciellt musikminne. Varje element i listan innehåller tonlängd och tonhöjd. I modulen har man en räknare som räknar ner tiden för den nuvarande tonen och en annan modul som bara tar in 5 bitar (32 toner ≈ 2,5 oktaver) som säger vilken ton som ska spelas just nu. De 5 bitarna används för att slå upp i en tabell över hur långa pulserna ska vara för den tonen. Sen växlar den bara mellan att skicka ut 0 och 1 på högtalaren.



Figur 2: Blockschema över musikenheten

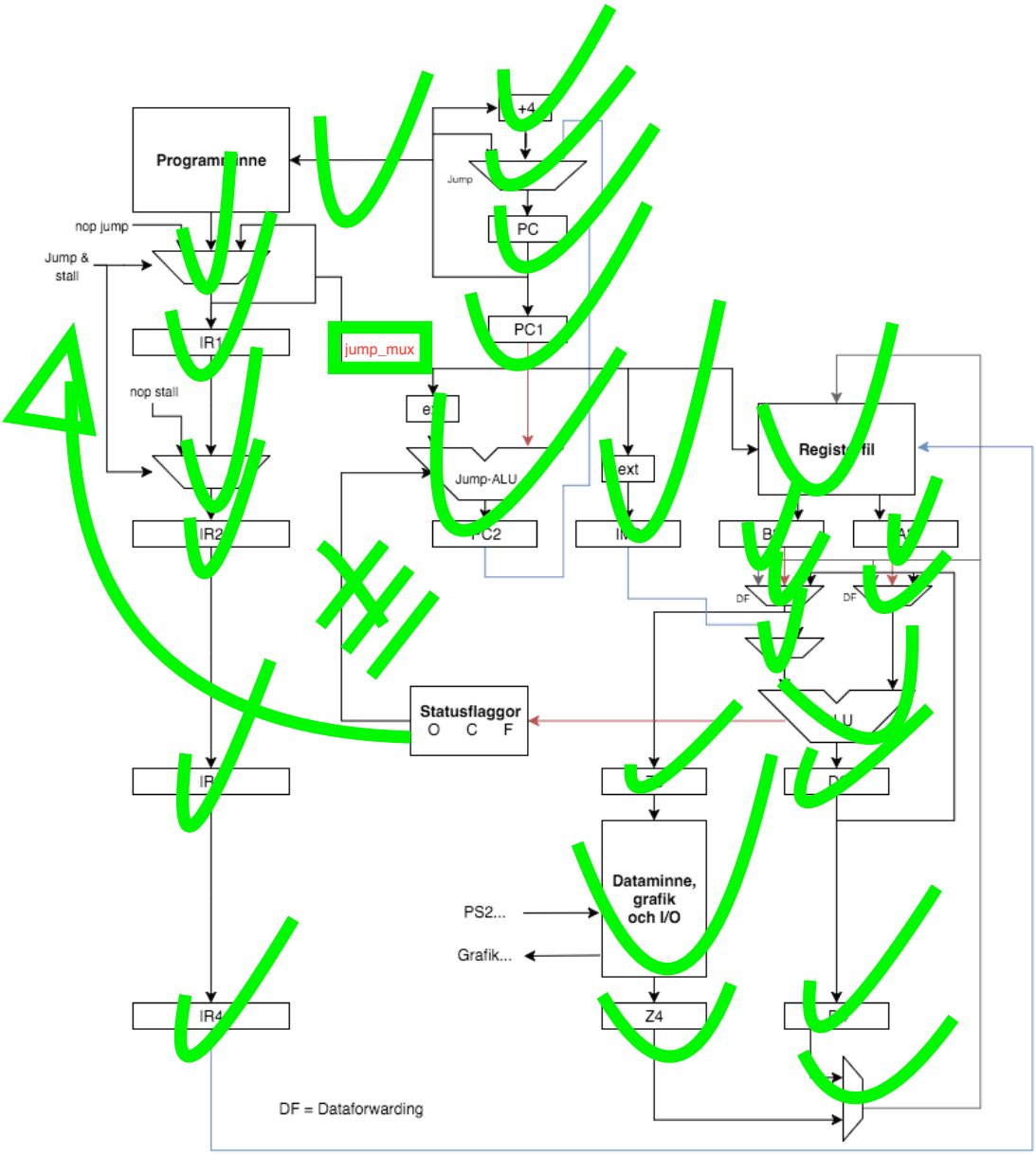


Figur 3: Blockschema över PS2-enheten



Figur 4: Blockschema över grafikenheten

Datorn använder minnesmappad I/O för att läsa från tangentbordet och skicka data till grafikenheten.



Figur 5: Blockschema över CPU:n

5 Milstolpe

Efter halva projektet ska processorn kunna exekvera alla instruktioner och grafiken ska kunna rita ut tiles på VGA-skärmen. Detta innebär att tileminnet är klart och fullt fungerande.