

# Designskiss

## Synth

Grupp 34

Version 0.2

Joacim Stålberg	joast229
Mattias Ulmstedt	matul773
Viktor Ringdahl	vikri500

# 1. Inledning

Denna designspecifikation beskriver en subtraktiv digital monosynth.

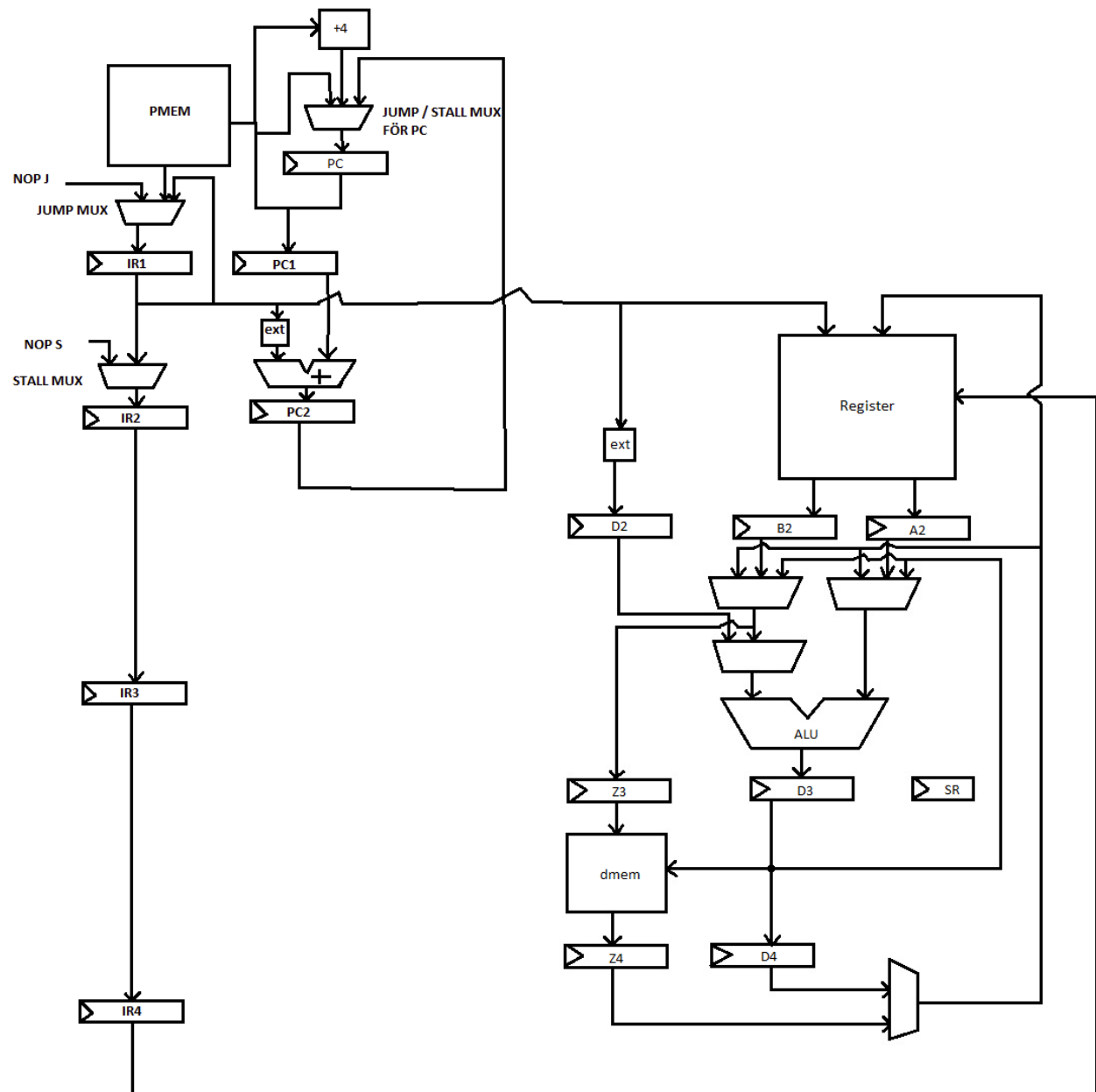
## 2. Analys

För att börja så måste man följa specifikationer för existerande hårdvara. Här följer relevanta observationer:

- LCD skärmen är 480x272 pixlar samt att från skärmen så skickas det tolv bitar med MSB först följt av tre nollor. In till touchpaden så ska 24 bitar skickas in för att bestämma RGB, dock upprepas samma värde för alla bitar då skärmen ska vara enfärgad.
- Stereo Audio ska ha in 16 bitar för att bestämma vad för ljud som ska skickas ut. För att få rätt förhållande på MCLK/LRCK som ska vara 256 så måste LRCK vara 44.1KHz.
- LCD-skärmen har en pixelklocka som ska vara minst 7 MHz och max 12MHz.
- MIDI keyboard: Ett midimeddelande består av en byte status och beroende på meddelande en eller två bytes data, men i denna synth är det bara meddelanden med två databytes som är intressanta. Överföringshastigheten bör vara 31250 Hz.
- Blockramet ska användas för att spara vågformer, tabeller (för uträkning av sinus med CORDIC) samt karaktärer som ska visas på skärmen. Tillgängligt är 576Kbits. För exempelvis 8 vågformer krävs ca 8000 bitar av dem (med 64 steg st), tabellerna kommer ta upp något i omkring 1000. Varje karaktär (8x8) tar då upp 64 bitar, och antalet karaktärer blir omkring 30, alltså ca 2000 bitar. Sedan behövs förstås skärmen kunna representeras av tiles, 32 möjliga karaktärer ger då 10200 bitar för hela skärmen, men det är mycket möjligt att varje sprites pixel blir 2x2 pixlar på skärmen och då skunker det ytterligare, Detta gör att användningen av block-RAMet bör hamna under det tillgängliga.
- Synthen har ungefär 2300 CK på sig att räkna ut varje sample. Lite av detta måste användas för att räkna ut mindre tidskritiska saker.

### 3. Konstruktion

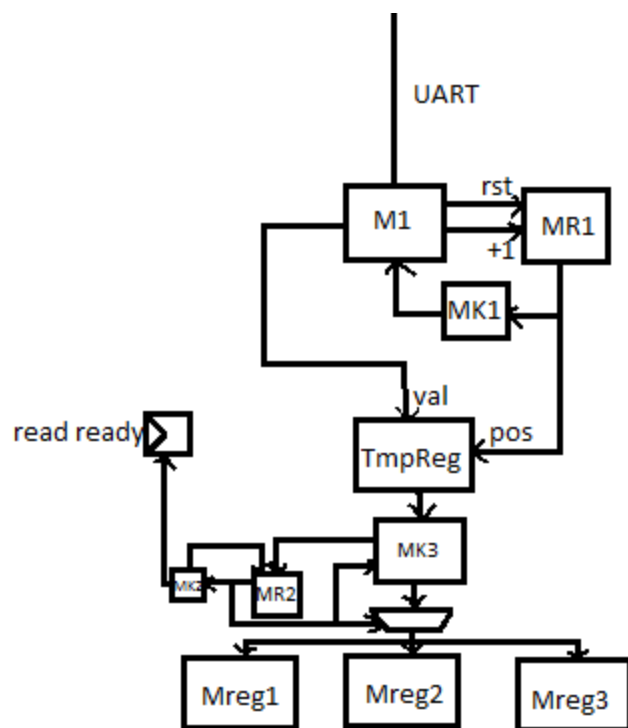
#### 3.1 Blockschema för CPU



Figur 1. Blockschema för CPU

Relativt rätt-fram pipelinad dator där CPU:n beskrivs övergripande. De mer specialiserade komponenterna beskrivs nedan.

### 3.2 MIDI över UART

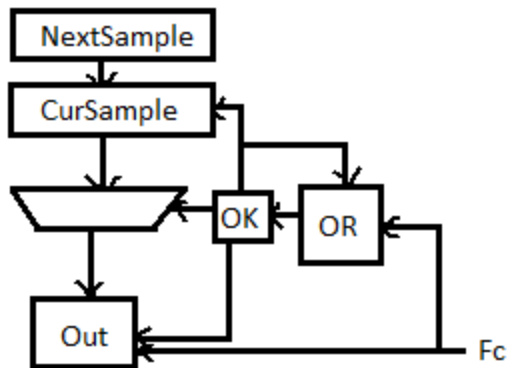


Figur 2. Blockschema för inläsning av keyboard.

- M1: Läser av UART:en och väntar på startbit.  
MR1: Räknare som räknar upp till en hel UART-byte.  
MK1: Kollar om vi har räknat klart och säger till M1 att vi är klara.  
TmpReg: Temporärt register för den senast inlästa byten.  
MK3: Läser av TmpReg och kollar om det är data som vi är intresserade av, och aktiverar isåfall räknaren MR2. Om MR2 inte är 0 är vi alltid intresserade av datan.  
MR2: Räknare som styr MUX:en för till vilket register vi vill spara TmpReg.  
MK2: Kollar om vi har räknat upp till 3, dvs läst in hela meddelandet (3 bytes). Resetar då räknaren samt sätter en flagga som säger att data finns att läsa.  
Mreg1-3: Register dit vi läser in MIDI-meddelandena.

Mreg1-3 är en del av registerblocket i CPU:n och read ready är en statusflagga. Se dokumentation [7] om MIDI för mer information om varför vi inte är intresserade av all information som skickas över MIDI.

### 3.3 Stereo-out



Figur 3. Blockschema för hanteringen av ljud-output.

NextSample: Buffer för den nästkommande sample:n.

CurSample: Den nuvarande sample:n som skickas till Out.

OR: Räkare som håller koll på hur många bitar som har skickats till Out. Räknar till dubbla sample-bitlängden.

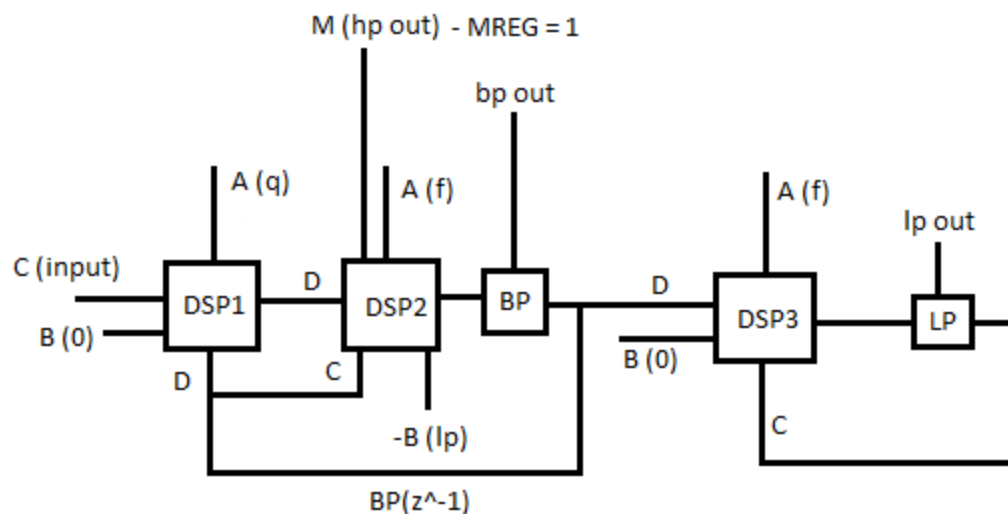
OK: Plockar ut översta biten som är höger/vänster samt de undre bitarna som väljer vilken bit i CurSample som skickas ut. Reset:ar även OR när den har räknat klart.

Fc: Samplingsfrekvensen \* 16 (eventuellt dubbla).

Out: PmodI2S-komponenten.

NextSample är ett register i CPU:n.

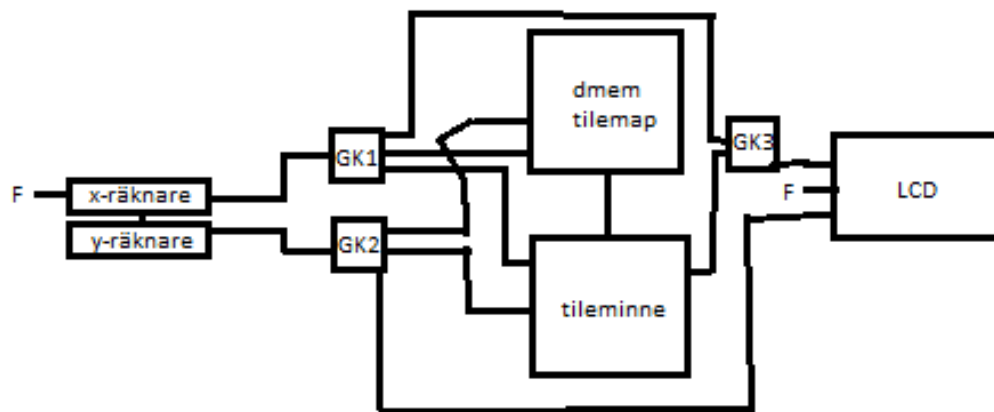
### 3.4 Filtret



Figur 4. Blockschemata för SVF-filtret.

Input och output är register i CPU:n. Filtret består av tre st DSP-slice:ar. Mer information om hur det fungerar i detalj finns att läsa bland referenserna ([5] och [6]).

### 3.5 Touch/LCD-skärm



Figur 5. Blockschemata för output till LCD-skärmen.

Räknare: Stegar igenom alla pixlar som ska ritas ut. Räknar lite högre än skärmens pixelbredd/höjd så att vi får lite delay mellan varje frame.

dmemo: Innehåller tilemap.

tileminne: Statiskt minne för hur alla tiles ser ut.

GK1: Plockar ut de 3 LSB och skickar in till tileminnet för att välja vilken del av tilen

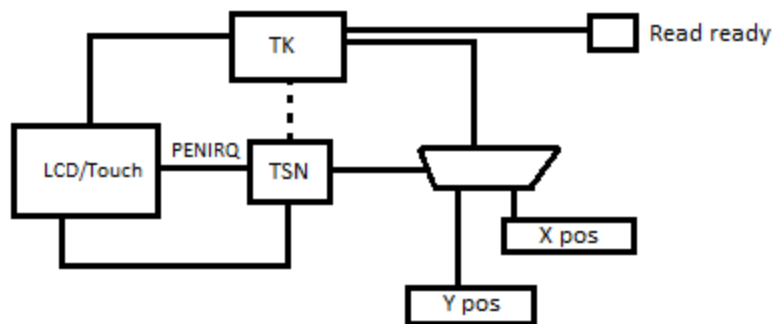
som ska ritas ut, och de 6 efterföljande bitarna och skickar in till tilemapen för att välja vilken tile som håller på att ritas ut. Till sist skickas även MSB in till GK3.

GK2: Samma som GK1 med skillnaden att MSB skickas in till LCD:ns DE.

GK3: Ser till att inget data skickas ut om x-räknaren räknat över LCD-skärmens bredd.

F: Skärmens uppdateringsfrekvens.

Dmem är en del i CPU:ns dmem.



Figur 6. Blockschema för input till LCD-skärmen.

TSN: Avläser om PENIRQ är aktiv och frågar isåfall efter positionen.

TK: Får in positionen och konverterar det till en 8-bits position. Sätter flaggan read ready när data är redo att hämtas.

X/Y-pos: Laddas med den konverterade positionen från TK.

X/Y-pos är register i CPU:n och read ready är en statusflagga.

## 4. Komponentlista

- Nexys3
- LCD touchskärm VmodTFT
- Stereo Audio PmodI2S
- MIDI-keyboard

## 5. Plan B

Om projektet visar sig bli allt för stort finns det möjlighet att skala bort vissa delar och ändå kunna demonstrera en synth. Om prestandan visar sig bli ett problem kan antalet voices sänkas (till en i allra värsta fall) samt så kan den interna beräkningarna sänkas till 8-bitar (som SID-chippet till exempel).

I värsta fall kan filtret tas bort om det visar sig att DSP-slice:arna inte vill fungera och att göra dem manuellt blir för ineffektivt. Om antalet voices sänks med finns det antagligen även möjlighet att göra filtret i mjukvaran istället.

## 6. Milstolpe

Efter halva projekttiden ska MIDI kunna avläsas och ljud spelas upp.



## Referenser

[1] Indiana Univeristy: Introduction to Computer Music: Volume One

[http://www.indiana.edu/~emusic/etext/MIDI/chapter3\\_MIDI4.shtml](http://www.indiana.edu/~emusic/etext/MIDI/chapter3_MIDI4.shtml)

[2] Digilent Nexys3 Reference Manual

[http://www.isy.liu.se/edu/kurs/TSEA83/kursmaterial/Litt/Nexys3\\_rm.pdf](http://www.isy.liu.se/edu/kurs/TSEA83/kursmaterial/Litt/Nexys3_rm.pdf)

[3] Digilent VmodTFT Reference Manual

[http://www.digilentinc.com/Data/Products/VMOD-TFT/VmodTFT\\_rm.pdf](http://www.digilentinc.com/Data/Products/VMOD-TFT/VmodTFT_rm.pdf)

[4] Cirrus Logic Stereo D/A Converter datasheet

[http://www.cirrus.com/en/pubs/proDatasheet/CS4344-45-48\\_F2.pdf](http://www.cirrus.com/en/pubs/proDatasheet/CS4344-45-48_F2.pdf)

[5] The digital state variable filter

<http://www.earlevel.com/main/2003/03/02/the-digital-state-variable-filter/>

[6] Spartan-6 FPGA DSP48A1 Slice User Guide

[http://www.xilinx.com/support/documentation/user\\_guides/ug389.pdf](http://www.xilinx.com/support/documentation/user_guides/ug389.pdf)

[7] MIDI Messages

<http://www.midi.org/techspecs/midimessages.php>