Linköpings universitet

Projekt rapport till digital konstruktion(TSEA43)

# Core War

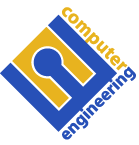


Jonas Hietala

Jesper Tingvall

Jizhi Li





Department of Electrical Enginnering

Linköpings universitet

Linköping 2012

## Sammanfattning

Beskrivning av CoreWars, vad vi har gjort etc?

## Innehållförteckning

## 1. Inledning

Projekt i kursen TSEA43 etc bla bla. Dokumentet är uppdelat bla bla

## 2. CoreWars '88

## 2.1 Beskrivning

Vad är det? Mål etc

Programmerarmodell?! Finns typ inte.

## 2.2 Instruktionsformat

**OP A B**

OP är en instruktion med operanderna A och B. Operanderna ska kunna adressera hela minnet. Hela instruktionen måste få plats på en enda rad.

Det finns totalt 4 adresseringsmoder och 11 instruktioner.

## 2.3 Adresseringsmoder

***Direct***

Operanden är en relativ offset som pekar på innehållet i en minnescell.

***Immediate***

Operanden är själva datat.

***Indirect***

Operanden är en relativ offset till en minnescell. Den cellens B operand använd som en relativ offset från den minnescellen.

***Predecrement-indirect***

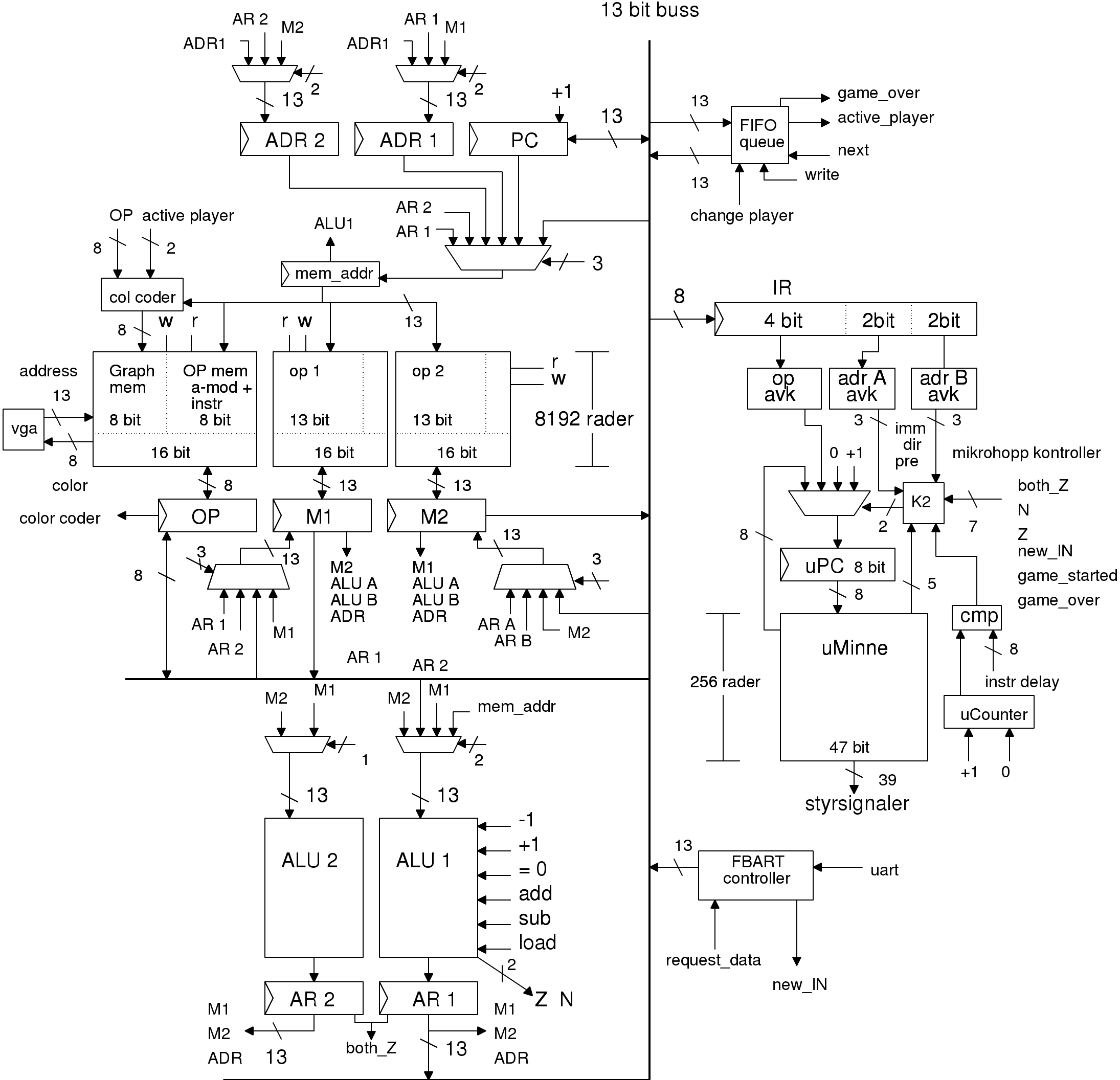
Operanden är en relativ offset till en minnescell. Cellens B operand minskas med ett, sätts tillbaks, och används sedan som en relativ offset till en ny minnescell.

## 3. Problemanalys ?? (Samma som DS? Behövs? Nåt annat här?)

Stuff...?

## 4. Blockschema

## 4.1 Huvudblockschema



Huvudblockschema + beskrivning

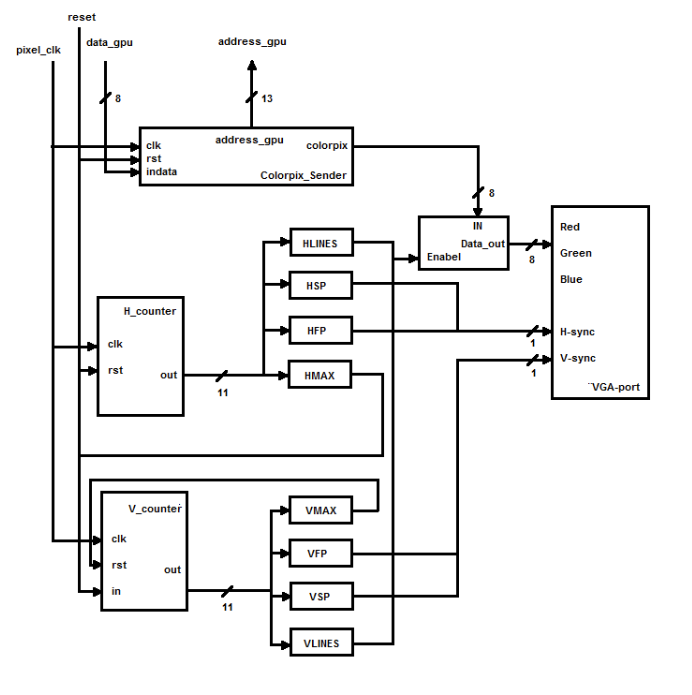
## 4.2 Fifo

fifo + beskrivning

## 4.3 Input

input + beskrivning

## 4.3 VGA/GPU/grafik (eller nåt)



VGA är delade upp till två delar: vga\_controller och pixelsender. Vga\_controller tar hand om att kontrollerar timing av signaler till VGA-port och pixelsender använder samma timing som vga\_controller samt hämta färg data urifrån huvudminne.

I vga\_controller finns det två räknare: h\_counter som räknar antalet horisontal pixel och v\_counter som räknar antalet vertikal pixel. Varje gång när h\_counter räknar upp till HMAX, dvs maximalt pixel vi har till skärm för varje rad, då kommer h\_counter själv nollställas och skicka en +1 insignal till v\_counter; v\_counter kommer att nollställas när den uppnå VMAX. (antalet pixel för varje kolumn)

HFP(slutpunkt till horisontal front porch), HSP(slutpunkt till horisontal synch pulse), VFP(slutpunkt till vertikal front porch), VSP(slutpunkt till vertikal synch pulse) kommer vi använda i vga\_controller. HFP kommer att aktiveras när h\_counter > HFP och skicka ’0’ signal till H-sync och HSP kommer att aktiveras när h\_counter > HSP eller h\_counter < HFP och skicka ’1’ signal till H-sync. VFP och VSP kommer att skicka sync signal till V-sync med på samma sätt.

VGA-port kommer endast ta emot färg data när h\_counter < HLINES(640 enligt upplösning vi valde) och v\_counter < VLINES(480 enligt upplösning vi valde) med hjälp av en enable signal från HLINE och VLINE.

Pixelsender använder samma timing och klock som vga\_controller, skcika ett 13 bitar address till minne, hämta 8 bitar data på detta address, upprepar och då skicka detta data till vga-porten endast när räknare in vga\_controller ligger inom display-ytan.

## 5. Hur fungerar vår grej (ehm)

## 2.1 Mikroprogrammering

Avkodning och exekverande av assemblyinstruktioner genomförs med hjälp av mikroprogrammering. T.T

## 2.2 Primärminne

Uppdelning av minne för att få plats eller nåt

## 2.3 ALU (? i 2.1?)

Awww crap

## 2.4 Fifo queue

Stuff

## 2.5 Input

Fbart dark magic here

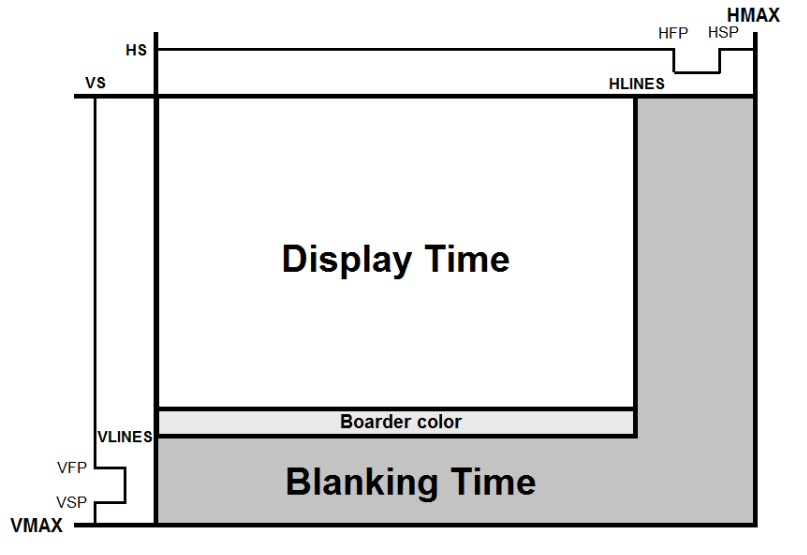
## 2.6 Output

PixelSender tar hand om address hämtning och färg kod sändning. För att alla data på skärm ska ses bra och tydligt ut, bestämde vi att visa varje data ska ha 5 pixel bredd och 7 pixel höjd. I så fall, kommer vi ha möjligt att visa 128 data per rad och vi behöver 7\*64 = 448 rader för att visa 213 = 8192 adresser. PixelSender skickar data till skärmen varje 5 klockor och upprepar detta för varje 128 data för 7 gånger, i så fall kan vi ha varje data med 5\*7 pixel storlek.

På vårt skärm finns det också en ytan som kallas ”border area”. I denna yta visar vi vilken spelare vinner CoreWar.

## 6. Beskrivning av hårdvara

## 6.1 VGA (Video Graphic Array)



När vga skickar pixel data till vga porten, kommer skärm inte ta emot och visa pixel data under hela tiden. Dessutom finns det ett speciallt timing till olika upplösningar med olika frekvenser, till upplösning 640x480 med frekvens 60Hz har vi följande timing enligt Digilent®.

* HMAX: 800
* VMAX: 525
* HLINES: 640
* VLINES: 480
* HFP: 648
* HSP: 744
* VFP: 482
* VSP: 484
* Clk: 25MHz

Orsaken till varför behöver vi har blanking time är att skärm använder strålen för att visa varje pixel och strålen flytter från vänster till höger och då flytter till nästa rad och upprepar denna process liksom hur h\_counter och v\_counter beter sig. Blanking time är nödvändigt eftersom mellan blanking time kommer strålen flytta sig från höger till vänster och under denna tid kan den inte visar någon pixel. Mellan front porch och back porch går sync signal ner och upp igen på grund av att det är sync signal som uppdaterar och bestämmer frekvens till skärmen.

På display ytan, kommer varje pixel uppdateras enligt 8 bitar färg som skärm har fått genom vga porten och på blank ytan, ska vga porten får ingen färg data alls, annars kommer skärm(strålen) visar detta färg när de flytter sig över skärmen.

## 6.2 Fbart

Hög per default! Dammit

## 6.3 FPGA

Hur fungerar den? Etc...

## 7. Beskrivning av assembler (eller som underrubrik till...?)

Humdidum?

## 7. ???

???

## 8. Diskussion

???

## Bilagor

Assembler format?

Utförligare instruktionsbeskrivningar?

Exempelwarriors?