|  |
| --- |
| AGSTU |
| HLS Ingenjörsjobb |
| ” Mandelbrot” |
|  |
| Olof Axelsson |
| **2025-03-01** |

Sammanfattning:

Ett projekt med HLS/Quartus/Eclipse där en komponent för beräkning av mandelbrotsmängden skapades. Denna komponent integrerades i Platform Designer i ett system, sedan användes Eclipse för att skriva C-kod som skriver ut fraktalen på VGA monitorn. C-koden kommunicerar med kompnentens register, skickar data till kompnenten som beräknas, svaret skickas tillbaka till C-koden ( NiosII/e ) och skrivs ut som grafik på en VGA monitor. Tidsåtgången för detta mäts med en IP komponent ( Timer\_e ), som sedan skrivs till konsolfönstret. För jämförelse HW/SW för denna beräkning så innehåller C-koden även kod för att räkna fram denna fraktal direkt i C-koden. Tidsvinsten var betydande mellan HLS-kompnenten / SW-komponent. 4 ( HW ) respektive 13 ( SW ) minuter.

Innehåll

[1 Kravspecifikation 3](#_Toc178233846)

[2.1 SDC fil 1 4](#_Toc178233847)

[2.2 SDC fil 2 4](#_Toc178233848)

[2.3 SDC fil 3 5](#_Toc178233849)

[3 Snabbaste frekvensen 6](#_Toc178233850)

[4 Slack 7](#_Toc178233851)

[5 För och nackdelar med de olika metoderna... 8](#_Toc178233852)

# Kravspecifikation

Tabell 1: Kravspecifikation från kund

|  |  |  |
| --- | --- | --- |
| **Krav id** | **Beskrivning** | **Utfört Ja/nej** |
| **Konstruktionskrav** | | |
| 1 | Konstruera minst en komponent med HLS-verktyget. | JA |
| 2 | Använd lämpliga optimeringsmetoder som har visats under träningen. | JA |
| 3 | Integrera komponenten i ett nios-system och validera på DE10-Lite utvecklingskort. | JA |
| 4 | Mainfunktionen och övriga funktioner bör ligga på separata kodfiler med tillhörande header-filer. | JA |
| 5 | TEIS regler och riktlinjer för C bör följas | JA |
| **Rapportkrav** | | |
| 6 | Skriv en vanlig rapport (.docx) som innehåller följande:   * Titelsida med namn, datum och en sammanfattning * Innehållsförteckning * Kravspecifikation, eget kapitel * Testprotokoll (enkelt) för verifiering/validering, eget kapitel * Konstruktionsbeskrivning, eget kapitel * Beskrivning av komponentens funktion och arkitektur (gränssnitt, lokalt minne, osv) * Verifiering, eget kapitel * Resultat från verifiering i co-simulation (ifyllt testprotokoll)   (Exekvera \*.exe efter kompilering för MAX10)   * (Eventuella skärmbilder från ModelSim) * Analys, eget kapitel * Analys av eventuella problem med kompilering, optimering, validering och övrig analys * Prestanda och area, eget kapitel * Beskriv hur komponenten har blivit optimerad * Resultat från de HLS-genererade rapporterna (resursanvändning, tidsanalys, initiation interval osv) före och efter optimering med procentuell förändring * Exekveringstider jämfört med motsvarande algoritm som körs i mjukvara * Validering, eget kapitel * Resultat från validering (ifyllt testprotokoll) * Eventuella bilder från VGA-display * Slutsats, eget kapitel * Erfarenheter och framtida förbättringar * Referenser, eget kapitel * Länk till YouTube presentationsvideo, se filmkrav | JA  JA  JA  JA  JA  JA  JA  JA  JA  JA  JA  JA  JA  JA  JA  Ja  JA |
| **Versionhanteringskrav** |
| 7 | Versionhantera projektet med Git och publicera på GitHub. |  |
| 8 | Repositoryt bör ha följande mappar och filer:   * Mappen *”hls”* * Kod-filer (en för varje komponent och testbänk) * Header-fil (för varje kod-fil) * Mappen ”*examples*” * Arkiverat Quartus-projekt * Kod-fil som använts för validering i NIOS II Eclipse. * Drivrutiner * .gitignore (exkludera NIOS-worskpace, Quartus projektmappar och filer som skapas vid kompilering med HLS-kompilatorn) * README.md (kort beskrivning av projektet) | JA  NEJ**\***  JA  JA  JA  JA  JA  JA |
| 9 | Lägg till ”*issues*” för eventuella förbättringar som kan göras (buggar, optimeringar, vidareutveckling mm..) |  |
| 10 | Om repositoryt är privat, skicka en collaborator-inbjudan till FPGA-HLS vid leverans av rapporten. (Repository settings / Collaborators) |  |
| **Filmkrav** |
| 11 | Projektet presenteras med en kort film (max 10 minuter) och ska minst innehålla följande:   * Översiktlig beskrivning av projektet på GitHub. * Beskrivning av C-koden som använts för att generera komponenten och testbänken. * Beskrivning av valideringsprocessen och en demonstration. * Visa eventuella ”*issues*” som lagts till.   Filmen ska läggas upp på YouTube (gärna med *keyword* TEIS)  **OBS;** Om du inte vill att filmen ska visas på Facebook med flera, skriv det efter länken. |  |
| **Leveranskrav** |
| 12 | Leveransen ska ske till plattformen Itslearning. Leveransen ska bestå av en zip-fil med namnet ”fornamn\_efternamn\_HLS\_ingenjorsjobb\_x.zip”. Där x motsvarar version. Rapporten inne i zip-filen ska levereras i Word-format med samma namn och versionsnummer som ovan. |  |
| 13 | Leveransen bedöms först ur teknisk synvinkel och komplettering kan göras maximalt 5 gånger om inte kunden (läraren) anser något annat. Därefter är det granskning av den tekniska rapporten och även här gäller det maximalt 5 kompletteringar för att få godkänt om inte kunden (rapportgranskaren) anser något annat. Om antalet kompletteringar överstiger dessa antal får den studerande ett helt nytt jobb att leverera till kund om läraren anser det motiverat. |  |

Punkt 8 ( Nej\* ) Jag förväxlade att man ska ha header filer osv. För HLS delen. Jag gjorde så istället för C-koden för Eclipse. Det var mer jobb så jag anser att jag kan det iaf.

# Testprotokoll

Tabell 2: Testprotokoll

|  |  |  |  |
| --- | --- | --- | --- |
| Räknare för Y uppdaterar ? | HW/SW komponent fungerar ? | Konsollfönster uppdaterar ? | Figur på VGA OK ?  ( Enkel Fraktal ) |
| JA | JA | JA | JA |
|  |  |  |  |

# Konstruktionsbeskrivning

Komponentens funktion är att beräkna om ett tal tillhör mandelbrotmängden eller inte.

Tillhör talet mandelbrotmängden visas detta på en VGA monitor genom att en pixel får vit färg, annars magenta. Detta upprepas för ett stort antal tal som representeras av X, Y på VGA monitorn. Vi sätter gränserna för beräkningen genom att välja WIDTH ( 640 ), HEIGHT ( 480 ) samt antalet iterationer att upprepa beräkningen innan vi anser att talet inte ingick i mandelbrotmängden ( 100 )

Komponentens arkitektur består av:

Kontrollregister

Statusregister

Indataregister Imaginärt tal

Indataregister Reelt tal

Returdata i form av antal utförda iterationer

I nedanstående figur kan vi se komponentens register för kommunikation.

Registren med offsetadress 0x38 och 0x40 är för framtida bruk och används ej i denna kod.

A screenshot of a computer

AI-generated content may be incorrect.

Figur 1: Mandlebrotkompnentens register.

# Verifiering

Verifiering av kompnenten i HLS miljön gjordes enkel enligt tabell nedan.

Tabell 3: Verifiering av komponent i HLS miljön

|  |  |  |  |
| --- | --- | --- | --- |
| Räknare för Y uppdaterar ? | HW/SW komponent fungerar ? | Konsollfönster uppdaterar ? | Figur ritas OK ?  ( Enkel Fraktal ) |
| JA | JA | JA | JA |
|  |  |  |  |

# Analys

Problem med kompilering var det. PATH för HLS krånglade med mera, sedan gick det inte att tilldela värden till flyttal för komponenten enligt ( imag float = 3.1415f; ) av någon anledning blev värdena korrupta då, lösningen var att använda pekare. Detta problem var beskrivet i vårat matereil från skolan så efter repetition av lektioner hittade jag det.

# Prestanda och area

Optimeringsmetoder

Jag använde kompilatordirektiven --fpc --fp-relaxed, jag provade även att använda

ac\_fixed<32,3,false> ac\_32\_3\_f; men det skapade bara problem senare I Eclipse.

Resultat

Figuren nedan visar komponenten före optimering

A screenshot of a computer

AI-generated content may be incorrect.

Figur 2: Mandlebrotkomponenten före optimering.

A screenshot of a computer

AI-generated content may be incorrect.

**Figur 3: Mandlebrotkomponenten efter optimering**.

Resursanvändningen före och efter optimering kan ses i nedanstpende tabell.

Tabell 4: Resursanvändning

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **1x clock fmax MHz** | **LCs** | **FFs** | **RAMs** | **DSPs** |
| Före opt. | 48.23 | 5506 | 2087 | 2 | 21 |
| Efter opt. | 66.76 | 1371 | 718 | 1 | 42 |
| Skillnad % | +38% | -75% | -65% | -50% | +100% |

Figuren nedan visar Initiation interval före optimering: = 7

A screenshot of a computer

AI-generated content may be incorrect.

**Figur 4: Mandlebrotkomponenten före optimering**.

Figuren nedan visar Initiation interval efter optimering: = 2

A screenshot of a computer

AI-generated content may be incorrect.

**Figur 5: Mandlebrotkomponenten efter optimering**.

Exekveringstider

Komponenten validerades och tiden mättes med en ”extended” timer som jag skapade i VHDL.

Mjukvarukomponent

12 minuter och 59 sekunder

Hårdvarukomponent

3 minuter och 53 sekunder

Så hårdvarukomponenten är 334% snabbare !!!

# Validering

Tabell 5: Valideringsprotokoll

|  |  |  |  |
| --- | --- | --- | --- |
| Räknare för Y uppdaterar ? | HW/SW komponent fungerar ? | Konsollfönster uppdaterar ? | Figur på VGA OK ?  ( Enkel Fraktal ) |
| JA | JA | JA | JA |
| Komponentens tidsbehov för 640x480x100 ( 100 = iterationer max ) pixels utrskrift på VGA monitorn. | | | |
| Mjukvara | 12m59s | Prestandan höjdes med 334% genom att hårdvaruaccelerera | |
| Hårdvara | 3m53s |

A screen shot of a computer

AI-generated content may be incorrect.

**Figur 6: Mandlebrotkomponentens utskrift på VGA monitor ( Fraktal )**.

Figuren ovan är skapad med en NiosII/e processor samt en VGA skärm ( 640 x 480 px. )

# Slutsats

Ja slutsatsen är att HLS kan vara väldigt användbart för att underlätta vid optimering av IP komponenter, särskilt om det gäller flyttalsmatematik. Såklart kan man optimera denna funktion mer och även lägga till fler register för 64 bitars beräkning vilket kommer ge möjlighet till mer detaljrika utskrifter.

# Referenser

Information om Mandelbrot

A screenshot of a computer

AI-generated content may be incorrect.

**Figur 7: Urklipp från Wikipedia, se länk nedan**

Två figurer är inklistrade på följande sida från Wikipedia för generell information om Mandelbrotmängden.

A screenshot of a computer screen

AI-generated content may be incorrect. A screenshot of a computer screen

AI-generated content may be incorrect.

**Figur 8: Urklipp från Wikipedia, se länk nedan**

[Mandelbrotmängden – Wikipedia](https://sv.wikipedia.org/wiki/Mandelbrotm%C3%A4ngden)

Länk till YouTube presentationsvideo