

Oblig 2 Mips simulator

Tarje Carlsen & Tristan Natvig

Ttc_96@hotmail.no

tristan.natvig@gmail.com

Github usernames : TarjeCarlsen & TristanK

Github repository : **Mips_oblig_2**

Dette er ikke en rapport, men mer en oppsummering av arbeidet vi har gjort og spørsmål vi har. Oppgaven er ikke ferdig og må gjentas på et andre forsøk. Det er generelle ting vi lurer på og som vi håper vi kan få en tilbakemelding på slik at vi kan utføre oppgaven korrekt og levere på nytt i andre forsøk.

Hva vi har gjort:

Vi har fått forståelse for hvordan en enkel syklus datapath er innenfor mips. Det er bygget opp av fem steg, IF, ID, EX, MEM, WB som avgjør hvor i syklusen programmet er. IF steget er for henting av informasjon fra memory, ID er for dekoding og register leses, EX er for å execute operasjoner og kalkulere adresser, MEM er for å få tilgang til minne, WB er for å skrive tilbake til register. Vi er fortsatt på å klare å løse stege for fetching hvor vi nå har laget en dekoder som etter mye egetlagede testkoder har fungert oppmot forventede resultat. Og vi har laget en kontroller som derfra skal skille ut fra den dekodete instruksjonen hvilke operasjoner som skal utføres. Dette er her vi jobber nå og vi holder ennå på med testing for å få denne til å fungere som forventet.

Vi har under oppgaven fått en del spørsmål som vi ønsker svar på for å utføre oppgaven videre.

1.

Hvordan er det vi skal koble sammen de forskjellige elementene? Vi har slitt med å forstå akkurat hvordan i CPUelements det er man setter verdier. Vi ser at det er en readInput, writeoutput og mye mer, men mangler forståelse av hvordan dette brukes i de forskjellige stepsene.

2.

Hvordan tester vi isolert et og et element? Vi har prøvd oss frem med ulike metoder for testing, men ikke helt forstått hvordan vi kan teste for eksempel at registerfile fungerer som den skal. Vi har prøvd å sette opp enkle funksjoner innenfor filen for så å teste at vi klarer å endre verdiene på registerene, men ettersom dette er python dictionary og ikke faktiske registre gir det oss lite tilbakemelding på om vi var på rett spor eller ikke.

3.

Når vi har jobbet med datapathen har vi prøvd å jobbe utifra hvordan den går. Vi begynner med program counter som da skal gi ut et multiple av 4, videre til registerfile. Skal vi da før vi kommer til registerfile gjøre dekoding av det vi får inn for så å skille op code, type, adresse osv og isåfall hvordan får vi påvirket det som kommer inn ved bruk av vår dekoding? Skal vi readinput i dekoderen for å returnere med writeoutput, eller gjøres dette på en annen måte?

4.

Mesteparten av tiden vi har brukt på oppgaven har blitt benyttet i runSimulator filen. Her, etter vår forståelse er det selve simuleringen av mipsen skal kjøre gjennom. Derfor har vi tenkt at vi da skal hente pc valuen. Dekode det vi får inn. Sende dette til registermodulen. Sende dette videre til ALU som derfra går videre til memory modulen. Er dette riktig tenkt eller burde vi gå fram på en annen måte?

Nåværende kode som blir sendt inn er ikke ryddet eller gjort klar for å gå over. Dette fordi den fortsatt arbeides på og oppgaven uansett må leveres på nytt. Håper besvarelsen av oppgaven går greit som et failed første forsøk.