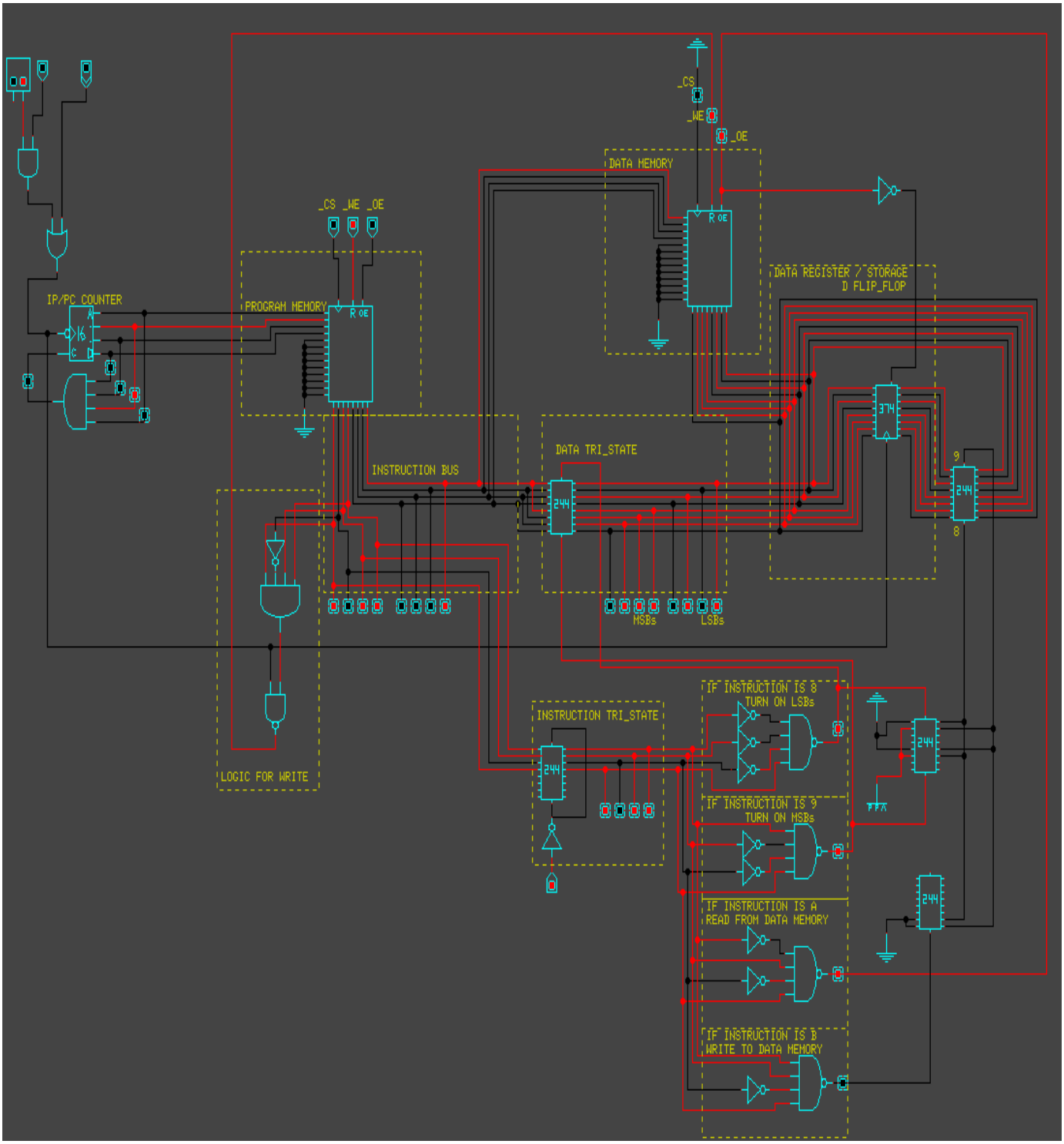
INF2270 – Oblig2
Aryan Esfandiari (aryane)

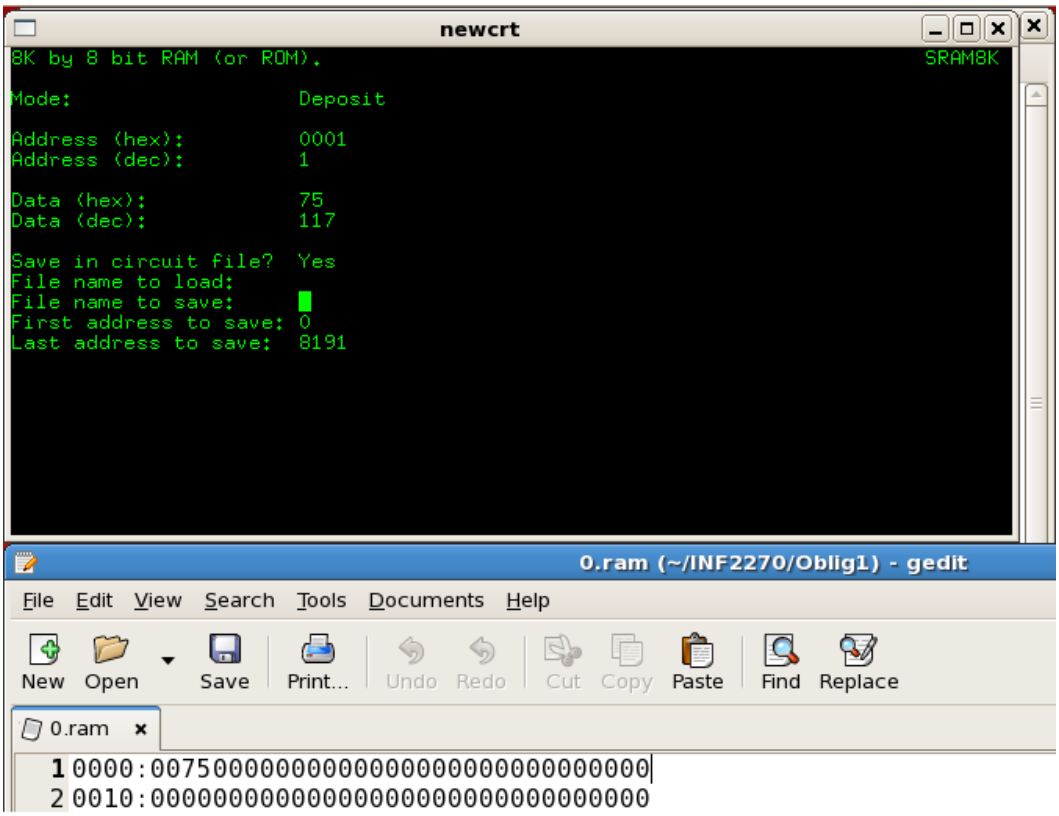
På obligen skulle vi lage en egen definert CPU, den er 8-bit.
Av 8-bit er de første 4 instruksjoner (MSBs) og de 4 andre data (LSBs).
De fire instruksjonene er definer, og det er å lese instruksjon, lese data, lese fra minne og skrive til minne. Og da må man analysere hva slags instruksjon vi har deretter behandle data.
Vi bruker også i tillegg en 4 bit teller som gir oss adresser til 16, men vi bruker kun halvparten, og det er pga hver gang vi leser, for vi 8-bit samtidig altså 2byte om gangen.
I programmet har vi 2 minner (SRAM8K), og de brukes mye.
Den ene er program minne, som programmet som vi skal utføre er i, og den skal vi bare lese.
Den andre som vi kaller for data minne er en midlertidig minne, som vi skal skrive og lese til.
Om vi skal lese eller skrive til data minnet avhenger av hva programmet vil.
En annen komponent som brukes mye her er 74244 som er egentlig en tri-state.
Siden vi bruker sammen bus men forskjellige logiker bruker vi tri-state komponent.
Ut av hva programmet vil skal vi kjøre forskjellige logikker til data minnet, og utgangen til hver logikk er koblet sammen tilslutt, for at det skal ikke bli konflikt kobler vi tri-state, da velger vi når skal hver enkelt logikk kjøres. Det er derfor vi bruker 74244.
En annen komponent som er brukt er 74374 som er egentlig en D-flip flop.
Den skal holde siste verdig og brukes her som en liten minne som skal lagre nåværende tilstand og forrige tilstand. CPU en vår gjør en enkel operasjon om gangen altså en byte.
Men lesing og skriving til minne er 2byte, altså resultat av 2 operasjoner.
Derfor må vi beholde forrige tilstand også i tillegg til nåværende.
Derfor bruker vi en liten tilstand maskin, som er D-flip flop.

Da har vi gått gjennom hver enkelt komponent men kort sagt om programmet:
1 – Les programmet fra progam minnet (som er også egentlig en .ram fil)
2 – Analyser innholdet, se hva den vil, del innholdet på instruksjoner og data
3 – Analyser instruksjonen og se hva den vil! (totalt 4 instruksjoner)
4 – Send også data videre til D-flip flopen som skal beholde forrige og nåværende data
5 – Analyzer om vi skal lese eller skrive til data minne, det blir også egen logikk

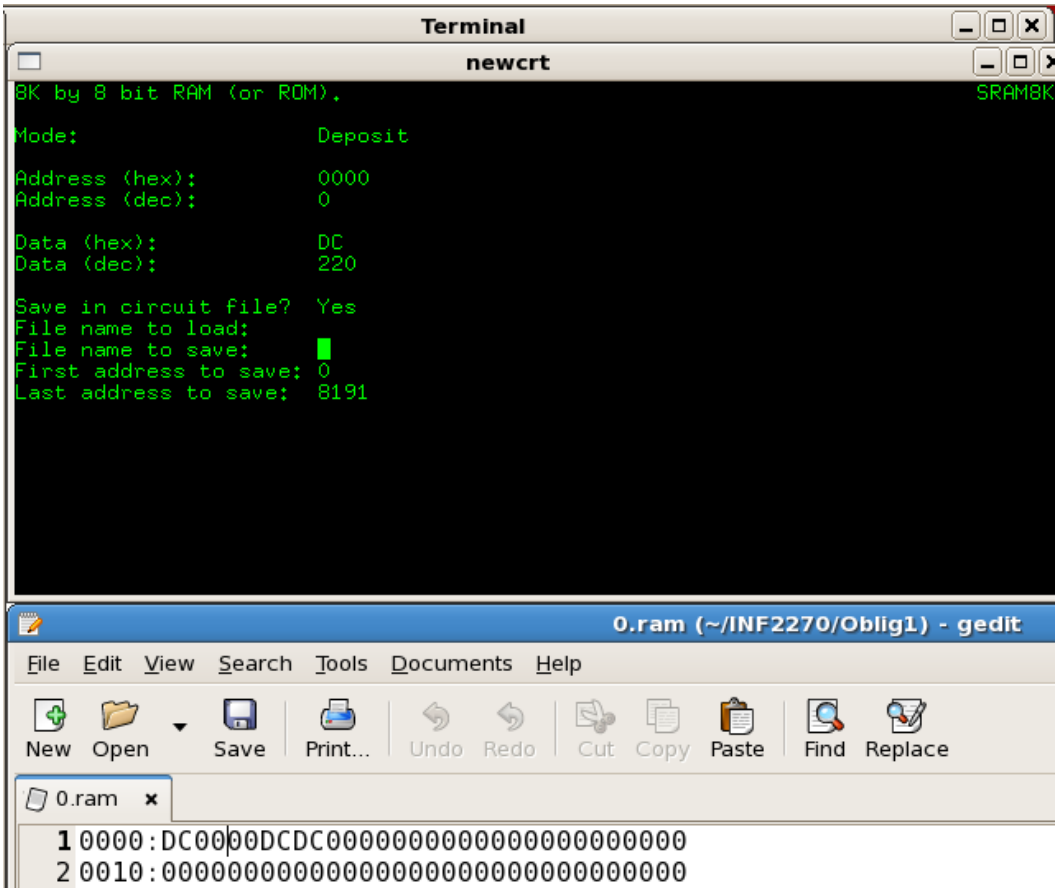
Og kort om lesing og skriving til minne (SRAM8K):
Denne minnet har 3 innganger for styring. _CS, _WE og _OE.
Understreket betyr at den er activ low, at den er automatisk høy.
_CS må være lav, for at den aktiviteter helle minnet.
_WE er write enable, som må være høy for lesing og lav for skriving.
_OE er output enable, som må være lav for lesing og høy for skriving.



Figur 1. Her er bildet er hele kretsen, som er laget ti digilog



Figur 2. Programmet 0000:8597B18194A10000 gir resultatet 0000:0075000000000000



Figur 3. Programmet 0000:DC0000DCDC000000 gir resultatet 0000:DCDC00DCDC000000