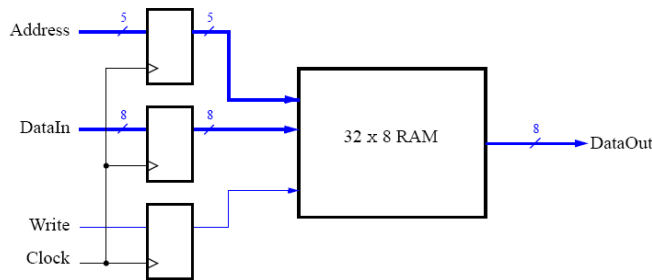


Innholdsfortegnelse:

Bruksanvisning for å lage en RAM med MegaWizard:	1
Hvordan legge inn en mif-file	6

F10_000 Bruksanvisning for å lage en RAM med MegaWizard:

Dersom du vil vite mer om LPM moduler, kan du se på *Using Library Modules in VHDL Designs*. Den ligger på CD-en som følger med kortet, i mappen: *DE2_System (E)\DE2_tutorials* og heter *tut_lpms_vhdl.pdf*. Den er en del av materialet som Altera har frigitt til *University Program section of Altera*. Ved å bruke den, lærer du hvordan du skal generere ønsket LPM modul.



(b) RAM implementation

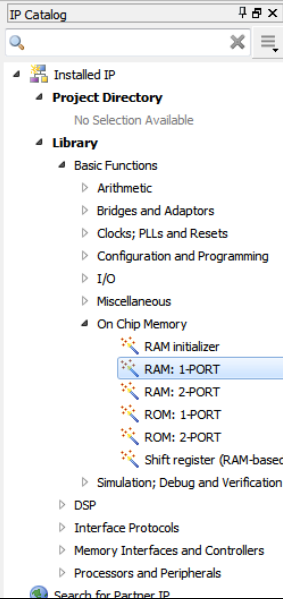
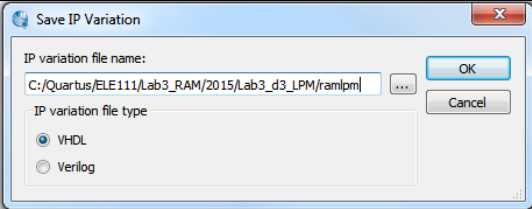
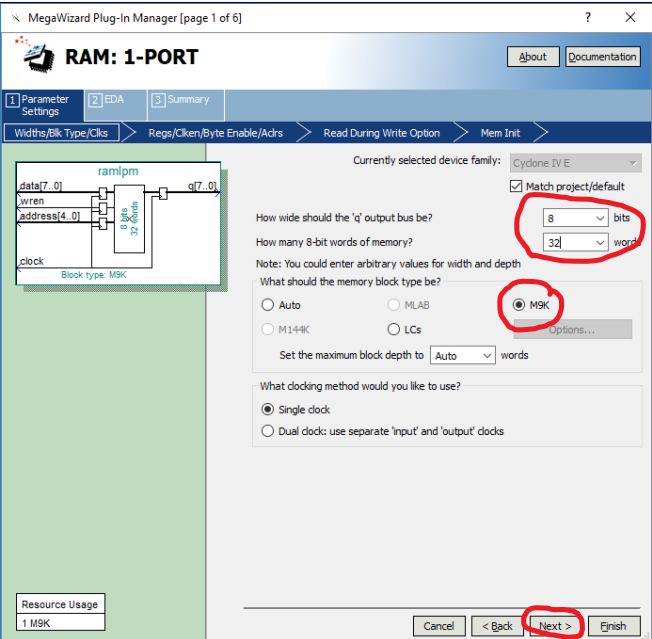
Figur 1: RAM med inn- og utsignal

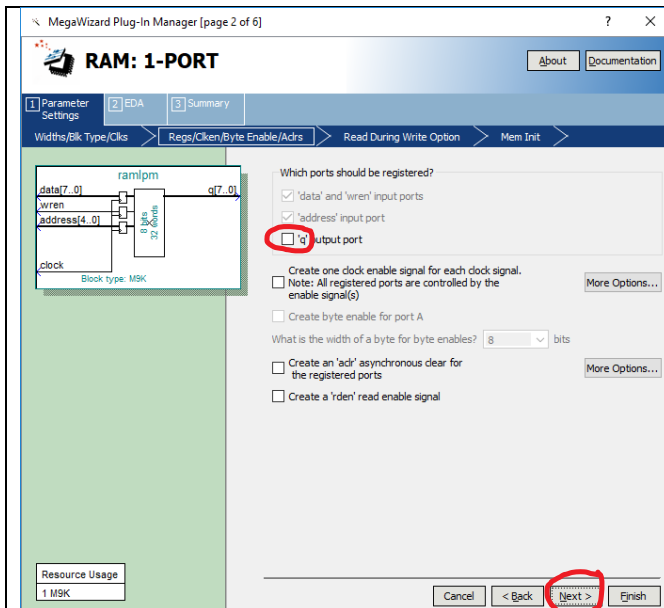
- 1) Lag et prosjekt som du kaller *ram_wizard_de*. Velg FPGA-en som står på DE2 kortet.
- 2) Opprett et prosjekt som du kaller *Ram_wizard_de2*
- 3) På Canvas under Lab3 ligger en file *Mal_Part3.vhd*. Lagre denne filen inn i dette prosjektet.
 - a. Endre navn slik at navn på entity, architecture og navn på toppnivå i Quartus-prosjektet er det samme.
- 4) Åpne filen. Ta analyse og syntese, om nødvendig, legg vhd-filen til i Quartus-prosjektet
 - a. Project -> Add/remove Files in project
- 5) Sett inn komponenten for 7-segment-dekoderen for tallene 0 til F.
- 6) Ta bort - -foran de siste linjene (tall4, tall5, tall6, tall7)
- 7) Få koden feilfri før du går videre.

Vi lager komponenten for RAM ved å bruke IP-biblioteket

Følg deretter det som står videre her:

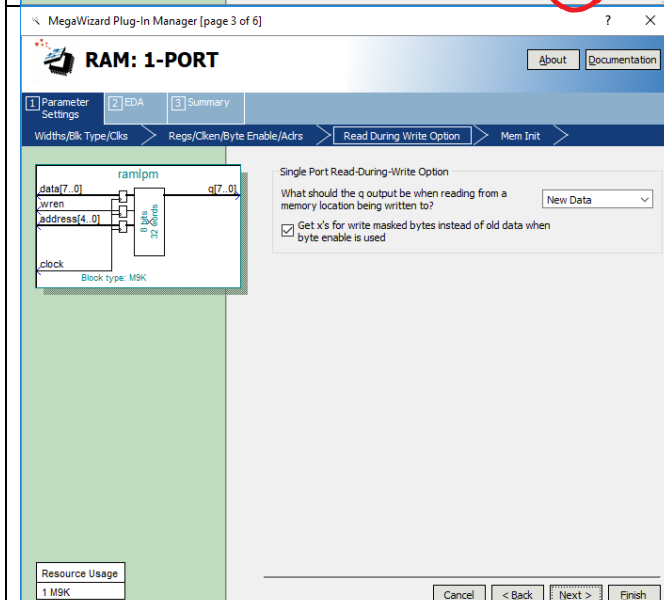
Velg Tools og klikk på: IP Catalog	Det kommer opp et vindu til høyre
Klikk på RAM: 1-port	

	
	<p>Dobbelklikk på <i>RAM: 1-PORT</i> under <i>On Chip Memory</i>.</p> <p>Merk av for VHDL, slik at filen blir lagret som en VHDL-fil</p> <p>Husk å gi ram-en et navn! Her har den fått navnet <i>ramlpm</i>.</p>
	<p>DE2-115:</p> <p>Vi velger å benytte en M9K- modul i FPGA-en</p> <p>Velg 8 bit output bus, 32 ord med minne.</p> <p>Trykk «Next >»</p>

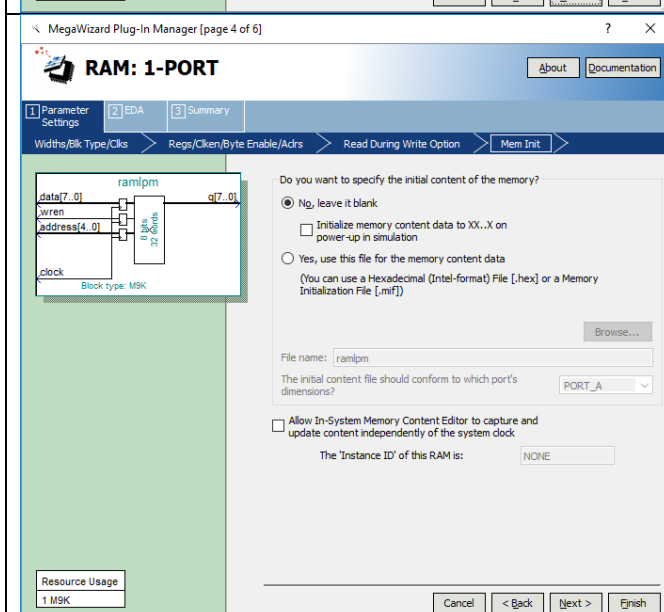


Vi ønsker ikke register på utgangene, så vi fjerner avmerkingen for det.

Trykk «Next >»



Trykk «Next ->»

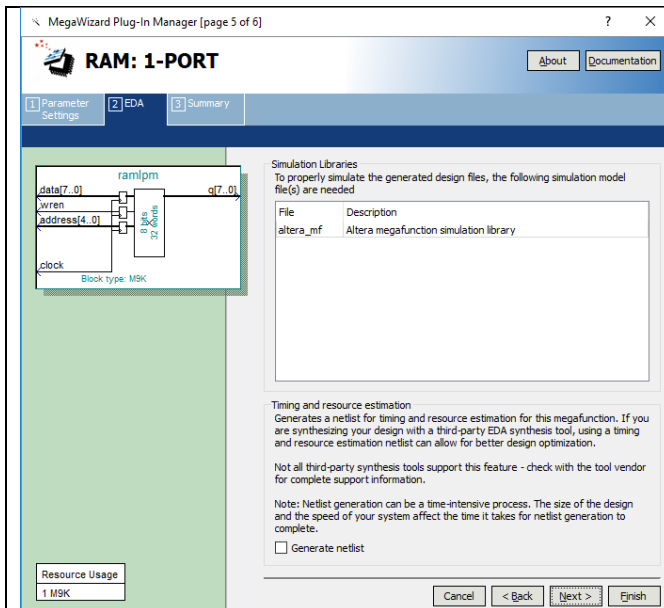


Vi ønsker ikke å bruke minne-file i første omgang, og merker derfor av av for NO. Minnefilen kan settes inn senere ved å dobbelklikke på:

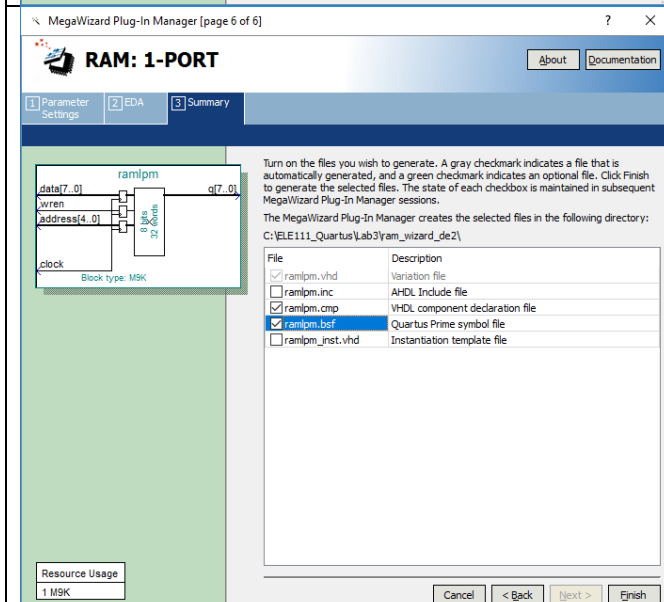


Dersom vi ønsker å bruke en minne-fil, merker vi av for YES. Filen kaller vi **ramlpm**. Det er viktig å huske dette navnet, for navnet MÅ brukes når minnefilen skal lagres.

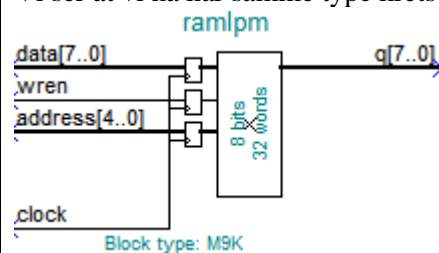
Vi lager ikke minnefile. (du kan prøve etterpå om du vil) (Prosjektet må kompiles før filen er på plass i hierarkiet)



Trykk «Next ->»

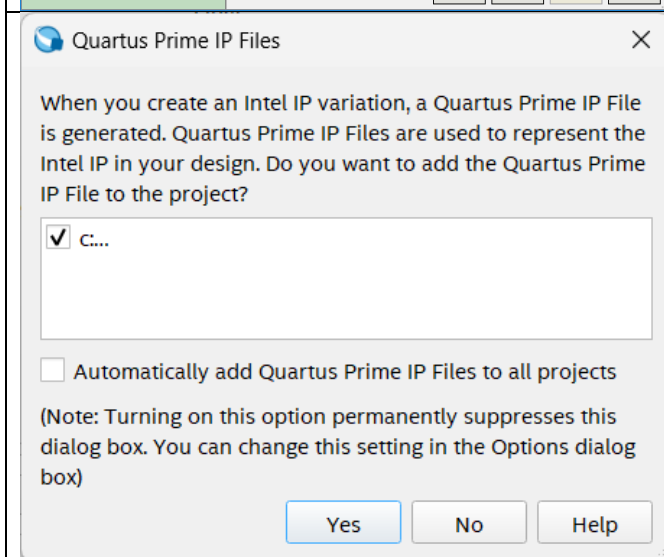


Finish!
Vi ser at vi nå har samme type krets som i figur1b



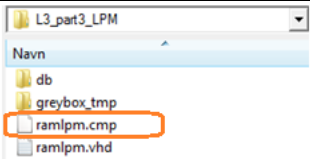
Kryss av for «ramlpm.cmp» og «ramlpm.bsf»

Velg «Finish»



Kryss av for c:..
Fjern evt kryss for «Automatically add...»

Trykk “Yes”

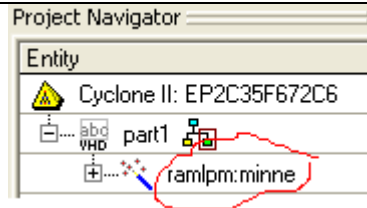
 <p>Åpn filen <i>ramlpm.cmp</i>. Komponenten <i>ramlpm</i> settes inn i vhdl koden, og kobles: Clock tl KEY(0), wren til SW(17)</p>	<pre>component ramlpm PORT (address : IN STD_LOGIC_VECTOR (4 DOWNTO 0); clock : IN STD_LOGIC := '1'; data : IN STD_LOGIC_VECTOR (7 DOWNTO 0); wren : IN STD_LOGIC ; q : OUT STD_LOGIC_VECTOR (7 DOWNTO 0)); end component;</pre>
--	--

Importer pinn-file. Kompiler. Last ned på kortet og test ut.

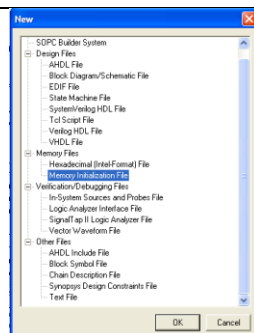
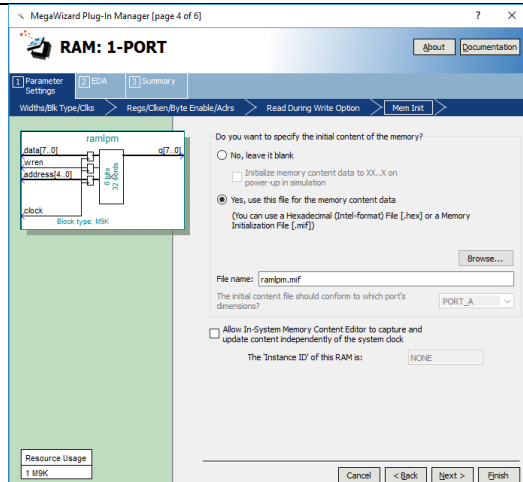
Hvordan legge inn en mif-file

Slik gjøres det dersom du ønsker å legge inn en mif-file (=Memory Initalization File)
(lag en kopi av prosjektet og prøv det ut der. Kall kopen part3_mif_NN) :

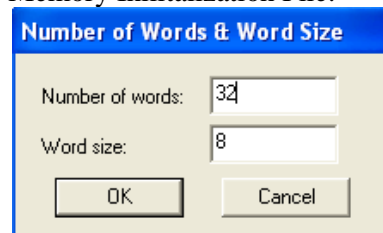
Minnefilen kan settes inn senere ved å dobbelklikke på:ramlpm:minne



Gå gjennom wizarden og hak av for Yes, og gi filen navnet **ramlpm.mif**.
Lukk wizarden.



For å lage mif-filen: Velg file,new og så Memory Initalization File.



32 ord
8 bit i hvert ord

Addr	+0	+1	+2	+3	+4	+5	+6	+7	+8	ASCII
0	00	00	00	00	00	00	00	00	00
9	00	00	00	00	00	00	00	00	00
18	00	00	00	00	00	00	00	00	00
27	00	00	00	00	00				

Vi kan nå fylle ut plassene i minnet. Kodene til 7-segmentet kan godt benyttes.

Addr	+0	+1	+2	+3	+4	+5	+6	+7	+8	
0	40	79	24	30	00	00	00	00	00	@
9	00	00	00	00	00	00	00	00	00	...
18	00	00	00	00	00	00	00	00	00	...
27	00	00	00	00	00					...

Fyll ut resten! Og lagr filen.

Husk at du allerede har bestemt hva den skal hete!
Jeg ga den navnet **ramlpm**, og må bruke det (**ramlpm.mif**)
når jeg lagrer filen.

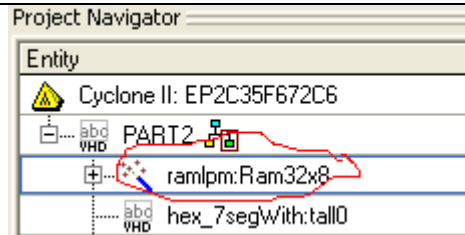
Forslag:

```
"1000000", --gfedcba, 40
"1111001", --, 79
"0100100", --2, 24
"0110000", --3, 30
"0011001", --4, 19
"0010010", --5, 12
"0000011", --6, 3
"1111000", --7, 78
"0000000", --8, 0
"0011000", --9, 18
"0001000", --A, 08
"0000011", --b, 02
"1000110", --C, 46
"0100001", --d, 21
"0000110", --E, 06
"0001110", --F, 0E
```

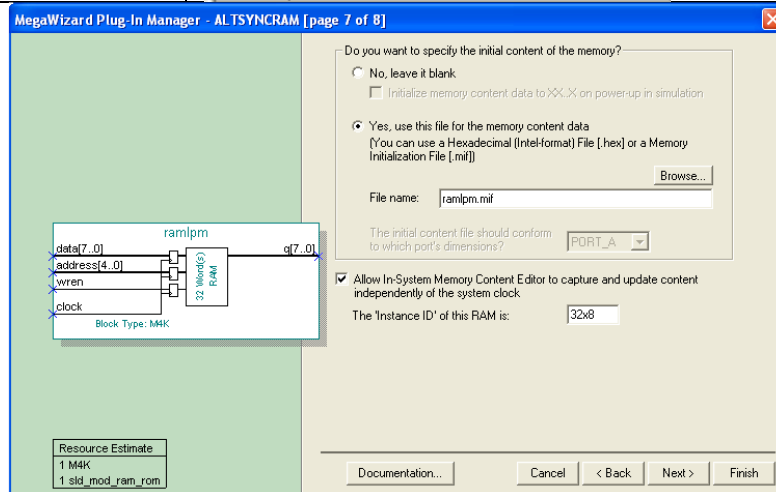
Bruk av In-System Memory Content Editor.

Utfør følgende:

- Lag en kopi av mappen NN-Part2 og endr navnet på mappen til NN_part2_MCE.
- Åpn prosjektet
- Vi skal nå gjøre endringer i minnemodulen. Klikk derfor på ram1pm i Project Navigator



- Bruk samme valg som i part1, men på siden der det blir angitt at minnefilen *ram1pm.mif* skal benyttes, hakes det også av for "Allow In-System Memory Content Editor to capture and update content independently of the system clock".



Dette valget tillater deg å bruke en egenskap ved Quartus II CAD som kalles *In-System Memory Content Editor*. Du vil da kunne se og gjøre forandringer på innholdet i RAM-modulen du har laget. Når du bruker dette verktøyet, kan du valgfritt spesifisere en 4-karakter 'Instance ID' som vil være navn på minnet.

- Fullfør de siste trinnene I wizardsen.

VHDL-kode der minne-modulen er med, har vi allerede. Bruk de samme brytere, LED-er og 7-segment som tidligere.

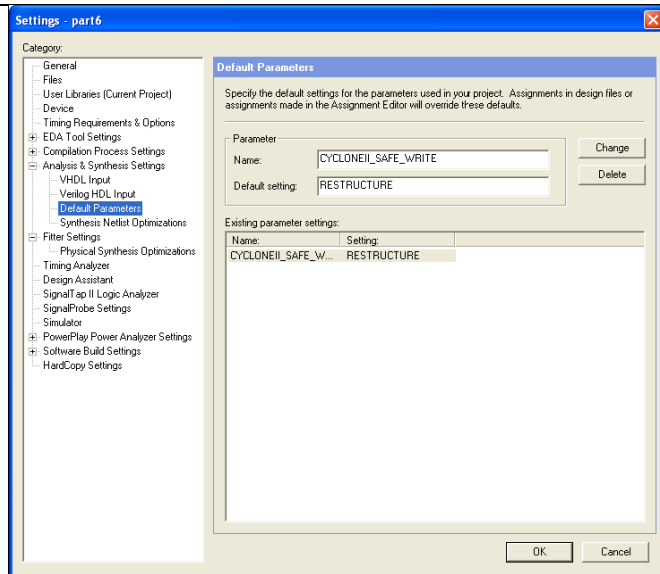
Før du kan bruke In-System Memory Content Editor tool, må du gjøre en tilordning til:

- Gå til *Assignments > Settings* og åpn vinduet *Default Parameters* under *Analysis and Synthesis Settings*.

- Skriv CYCLONEII SAFE WRITE som *parameter name*.

- Default setting settes til RESTRUCTURE. Dette parameteret åpner opp for at Quartus II syntese verktøyet gjør nødvendige modifikasjoner, slik at skrivning og lesing kan gjøres via *In-System Memory Content Editor tool*.

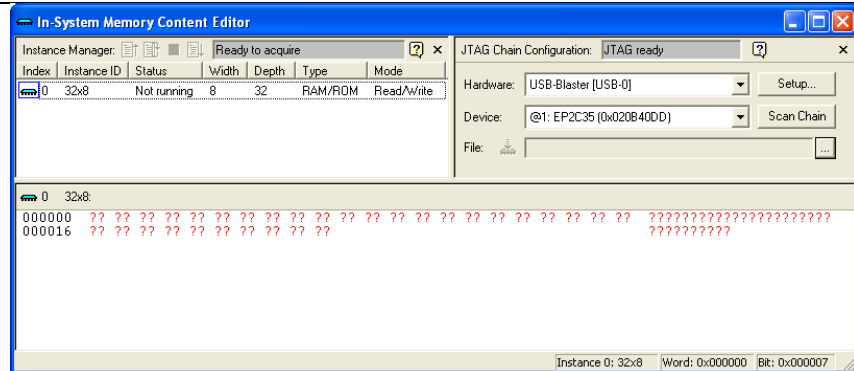
- Klikk OK.



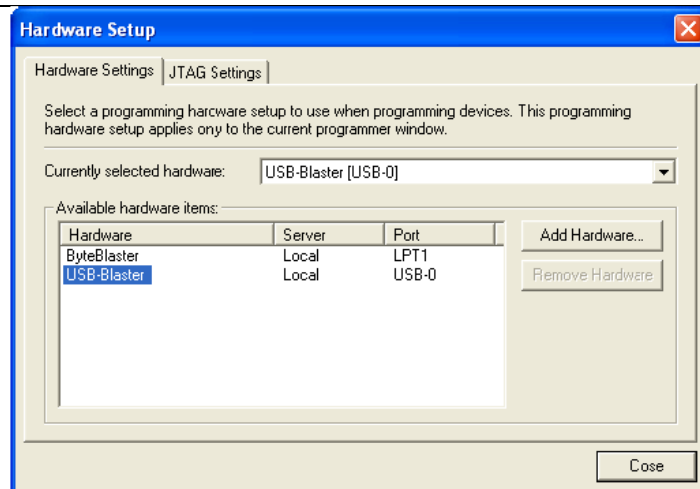
- Kompiler VHDL-koden og last kretsen ned på DE2-kortet. Test kretsen og forsikr deg om at les og skriv operasjonene fungerer som de skal.. Er det noen forskjell? Noter!

- Velg Tools > In-System Memory Content Editor. Dette vinduet kommer opp:

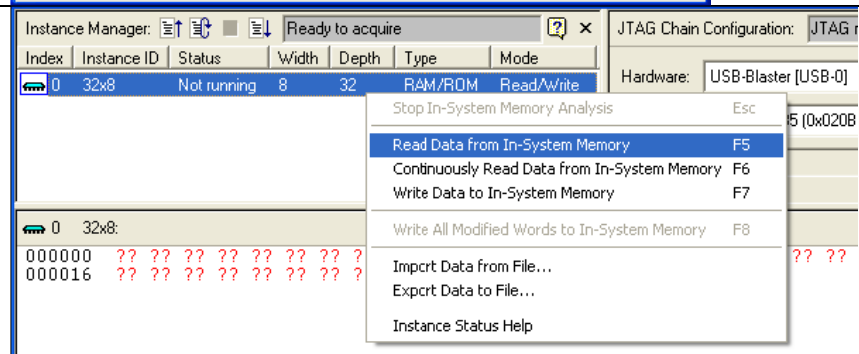
For å spesifisere forbindelsen til DE2 kortet, (dersom den ikke er i orden) klikk på Setup-ikonet på høyre side i vinduet.



Dette vinduet kommer opp:
Velg USB-Blaster.
Klikk på Close.



Instruksjoner for å bruke *In-System Memory Content Editor tool*, finnes i Quartus II Help.
En enkel operasjon er å høyreklikke på 32x8 minnet slik figuren viser.
Velg *Read Data from In-System Memory*. Dette gjør at innholdet i minnet vises i nederste del av vinduet. Du kan nå editere alle verdier som vises, ved å skrive over dem. for å skrive inn nye verdier i RAM, høyreklikk på 32x8 minnet og velg *Write Data to In-System Memory*.



Prøv deg fram og observer at data vises som de skal både på 7-segment displayene og i *In-System Memory Content Editor* vinduet.