راهنمای استفاده از نرم‌افزار Quartus

بر اساس نسخه 13.1

آشنایی با ابزار Mega Wizard

راهنمای پیش رو برای آموزش ابزار MegaWizard Plug-In Manager در نرم‌افزار Quartus نسخه ۱۳.۱ می‌باشد کلیه حقوق این اثر متعلق به گروه ICEEP دانشگاه تهران است. هر گونه تکثیر از این اثر منوط به اجازه‌ی کتبی پدیدآورندگان این راهنما در گروه ICEEP می‌باشد.

www.ICEEP.ir

روند بازبینی

نسخه	تاریخ	نسخه ISE
۱.۰	۹٤.۱.۲۲	۱۳.۱

www.ICEEP.ir

فهرست

فصل ۱: پیشگفتار ٤

فصل ۲: آشنایی با ابزار MegaWizard Plug-In Manager ٦

فصل ۳: مرور سریع بر نحوه ساخت پروژه جدید و ایجاد و کامپایل یک فایل HDL ۸

فصل ٤: نحوه استفاده از ابزار MegaWizard Plug-In Manager ۱۳

فصل ۵: نحوه استفاده از ابزار In-System Memory Content Editor ۲۲

Copyright © 2015 ICEEP, University of Tehran

فهرست اشکال

شکل (۳-۱) شمای کلی نرم‌افزار Quartus برای ایجاد یک پروژه جدید. ۹

شکل (۳-۲) تنظیم نام و محل ذخیره شدن پروژه جدید ۹

شکل (۳-۳) تعیین نام دستگاه مورد استفاده و نوع آن. ۱۰

شکل (۳-٤) مرحله نهایی تعریف یک پروژه جدید ۱۱

شکل (۳-۵) نحوه تعریف یک فایل HDL جهت کامپایل کردن و سنتز ۱۲

شکل (٤-۱) صفحه نخست استفاده از ابزار MegaWizard Plug-In Manager جهت استفاده از ابزارها و گیت‌های از پیش تعریف شده در Quartus ۱٤

شکل (٤-۲) نحوه استفاده از حافظه RAM موجود در Quartus و تعریف یک نمونه از آن با نام Megamem ۱۵

شکل (٤-۳) تعیین تعداد wordها و تعداد بیت‌های درون حافظه ۱٦

شکل (٤-٤) تنظیمات ریجیستر ۱۷

شکل (٤-۵) اضافه کردن فایل MIF ۱۸

شکل (٤-٦) اجازه ی تغییر در زمان runtime ۱۹

شکل (٤-۷) کتابخانه‌های شبیه سازی شده ۲۰

شکل (٤-۸) خلاصه ی تمام فایل‌های ساخته شده ۲۱

شکل (۵-۱) تنظیمات اولیه ۲۳

شکل (۵-۲) شمای کلی In system Memory Content Editor ۲٤

شکل (۵-۳) نمایش اطلاعات داخل حافظه ۲۵

Copyright © 2015 ICEEP, University of Tehran

شکل (۵-٤) تغییر در حافظه.....۲٦

فصل اول

پیشگفتار

در این راهنما به بررسی نحوه‌ی استفاده از ابزار MegaWizard Plug-In Manager در نرم‌افزار Quartus، جهت ساخت یک حافظه و استفاده از ابزار Run Time Memory Content Editor می‌پردازیم.

بدیهی است گفتار پیش رو برای آن دسته از افرادی می‌باشد که آشنایی ابتدایی با نرم‌افزار Quartus را دارند و با نحوه ایجاد یک پروژه جدید و سنتز کردن آن آشنایی ابتدایی دارند.

در ادامه ٤ فصل خواهیم داشت. در فصل ۲ توضیحات کلی در مورد ابزار MegaWizard Plug-In Manager را خواهیم دید. در فصل ۳ مروری سریع بر نحوه ایجاد یک پروژه جدید در Quartus خواهیم داشت. فصل ٤ به استفاده از ابزار MegaWizard Plug-In Manager جهت ساختن حافظه در Quartus می‌پردازد و در نهایت در فصل ۵ به نحوه استفاده از ابزار Run Time Memory Content Editor خواهیم پرداخت.

www.ICEEP.ir

فصل دوم

آشنایی با ابزار

MegaWizard Plug-In Manager

همانطور که پیچیدگی در مدارات افزایش می‌یابد ، استفاده از IP بلوک‌ها به عنوان یک روش طراحی بیشتر می‌شود. Altera در این زمینه یک سری توابع[1] قابل برنامه ریزی تولید نموده که برای معماری ابزارهای Altera استفاده می‌شود. استفاده از توابع در مقایسه با روش‌های دیگر زمان کمتری لازم دارد. همچنین این ابزار به ما کمک می‌کند که طراحی ما سنتز پذیری بالاتری داشته باشد.

Quartus II ۳ مدل megafunction را تولید می‌کند که برای ساخت حافظه استفاده می‌شود. شما می‌توانید با MegaWizard Plug-In Manager خصوصیات داخلی حافظه را تعیین نمایید. به عبارتی دیگر در نرم‌افزار Quartus به کمک MegaWizard Plug-In Manager می‌توانیم یک حافظه به سایز دلخواهمان را طراحی کنیم .

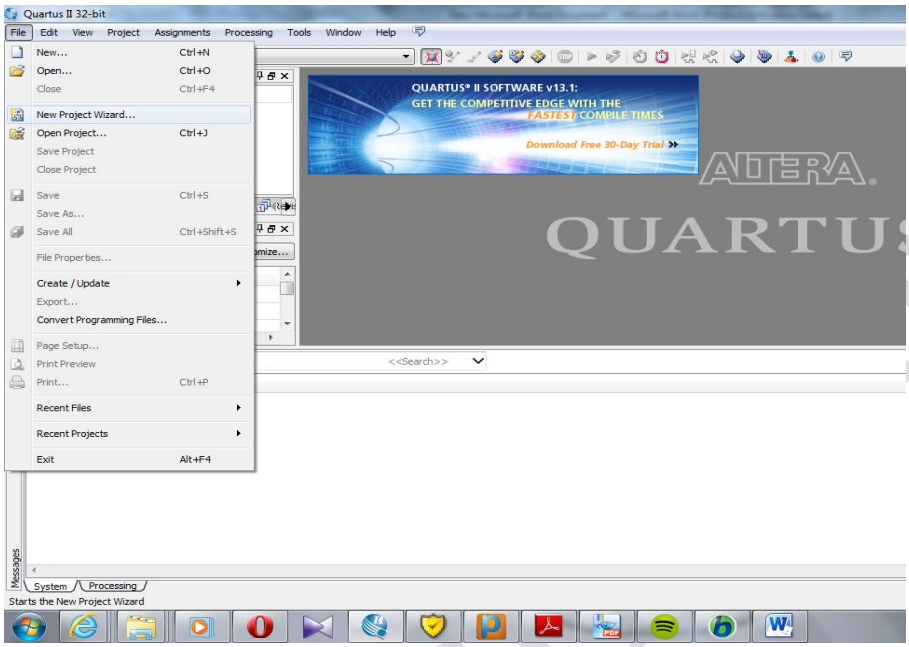
[1] Megafunction

www.ICEEP.ir

فصل سوم

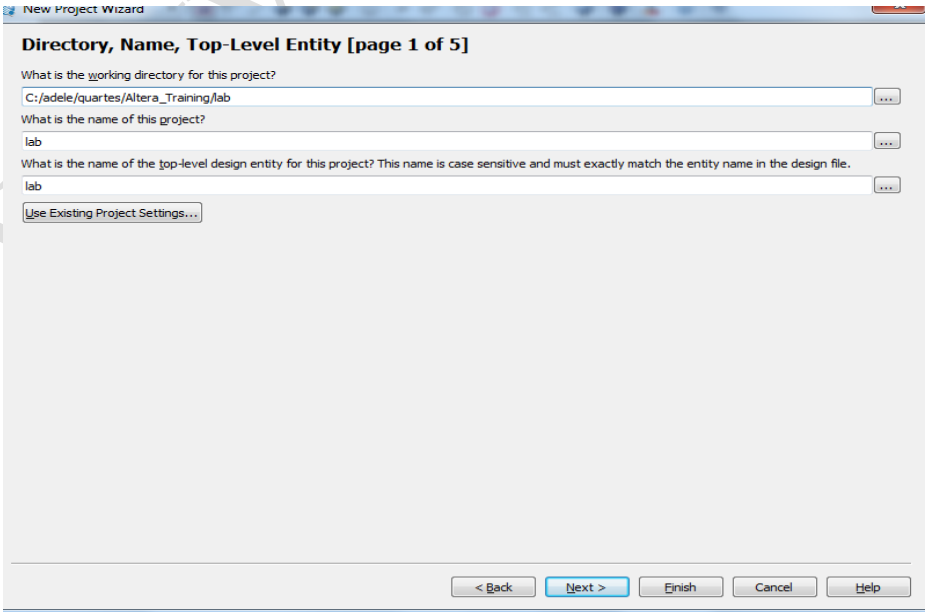
مرور سریع بر نحوه ساخت پروژه جدید و ایجاد و کامپایل یک فایل HDL

همان طور که در راهنمای شروع اولیه با Quartus II گفتیم برای شروع کار ابتدا باید پروژه‌ی مورد نظر را تعریف کنیم به منوی File می‌رویم و گزینه New project wizard را انتخاب می‌کنیم.



شکل (۳-۱) شمای کلی نرم‌افزار Quartus برای ایجاد یک پروژه جدید.

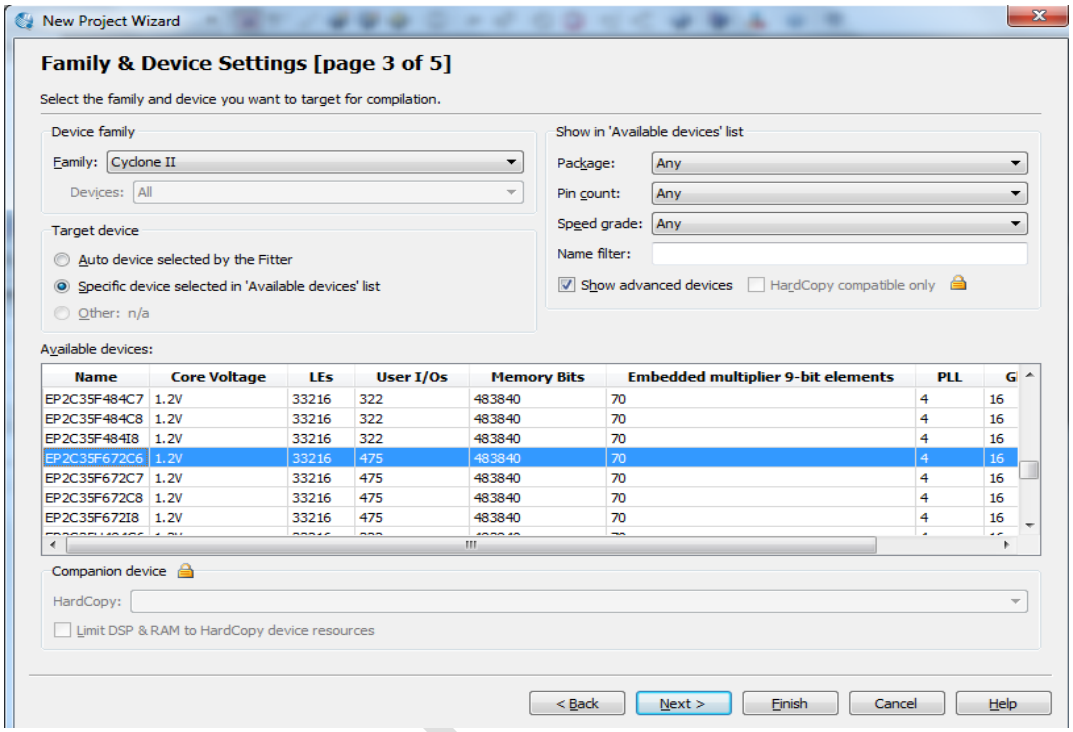
سپس در پنجره‌ی باز شده برای قسمت Introduction ، Next را انتخاب می‌کنیم و حال برای قسمت directory باید working Directory و Project Name را تعیین کنیم. برای مثال داریم:



شکل (۳-۲) تنظیم نام و محل ذخیره شدن پروژه جدید.

Copyright © 2015 ICEEP, University of Tehran

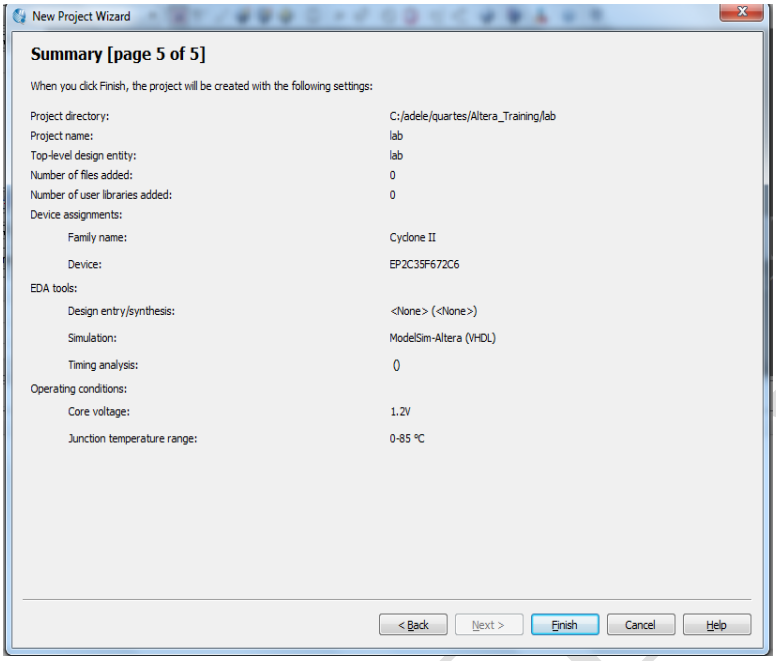
سپس Next را میزنیم ودر قسمت Add File همNext را می‌زنیم و در قسمت Family & Device در بخش Device Family ، Cyclone II را انتخاب می‌کنیم و در بخش Available Device، دستگاه خود را انتخاب می‌کنید که در اینجا EP2C35F672C6 است .



شکل (۳-۳) تعیین نام دستگاه مورد استفاده و نوع آن.

در ادامه کلید Next را فشار می‌دهیم و در صورتی که از ابزارهای دیگرEDA استفاده نشود دوباره کلید Next را فشار می‌دهیم و در انتها کلید Finish را فشار می‌دهیم .

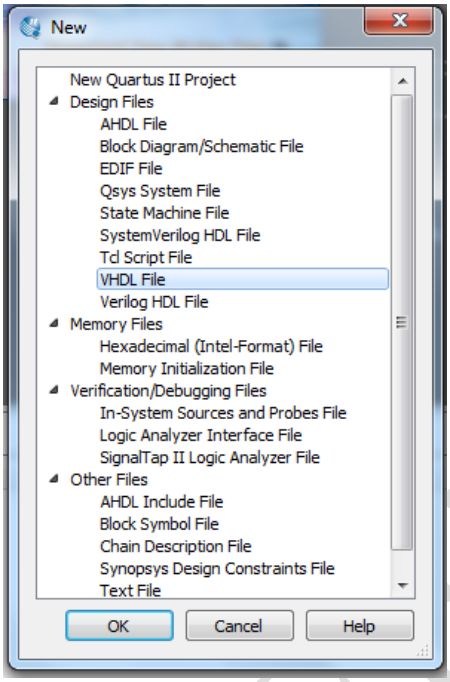
Copyright © 2015 ICEEP, University of Tehran



شکل (۳-۴) مرحله نهایی تعریف یک پروژه جدید.

برای تولید و کامپایل کردن یک فایل به صورت زیر عمل می‌کنیم. در زیر منوی File گزینه‌ی New را انتخاب می‌کنیم و در منوی باز شده گزینه‌ی VHDL File را انتخاب می‌کنیم و بر روی ok کلید می‌کنیم.

www.ICEEP.ir



شکل (۳-۵) نحوه تعریف یک فایل HDL جهت کامپایل کردن و سنتز.

حال پنجره‌ی سفیدی برای ما باز می‌شود که داخل آن می‌توانیم کد VHDL را وارد کنیم که در اینجا شما می‌توانید از کد خود استفاده کنید. سپس ازمنوی File گزینه‌ی Save را می‌زنیم و فایل را در همان پنجره Save می‌کنیم.

www.ICEEP.ir

فصل چهارم

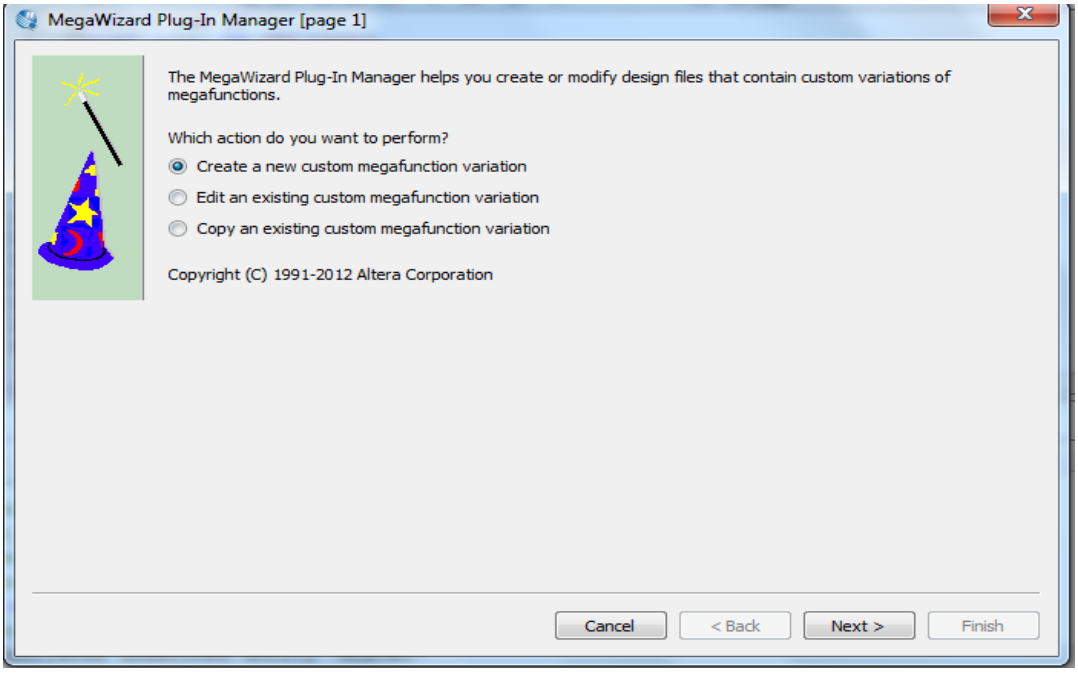
نحوه استفاده از

ابزار MegaWizard

Plug-In Manager

برای ساخت حافظه

در بخش قبل نحوه ایجاد یک پروژه جدید را به طور مختصر یاد گرفتیم. برای ساختن حافظه‌ها می‌توانستیم از ابزار MegaWizard® Plug-in Manager نیز استفاده کنیم . برای این کار ابتدا از منوی Tools، MegaWizard Plug-in Manager را انتخاب می‌کنیم. در ادامه صفحه ای مشابه شکل زیر ظاهر می‌شود.

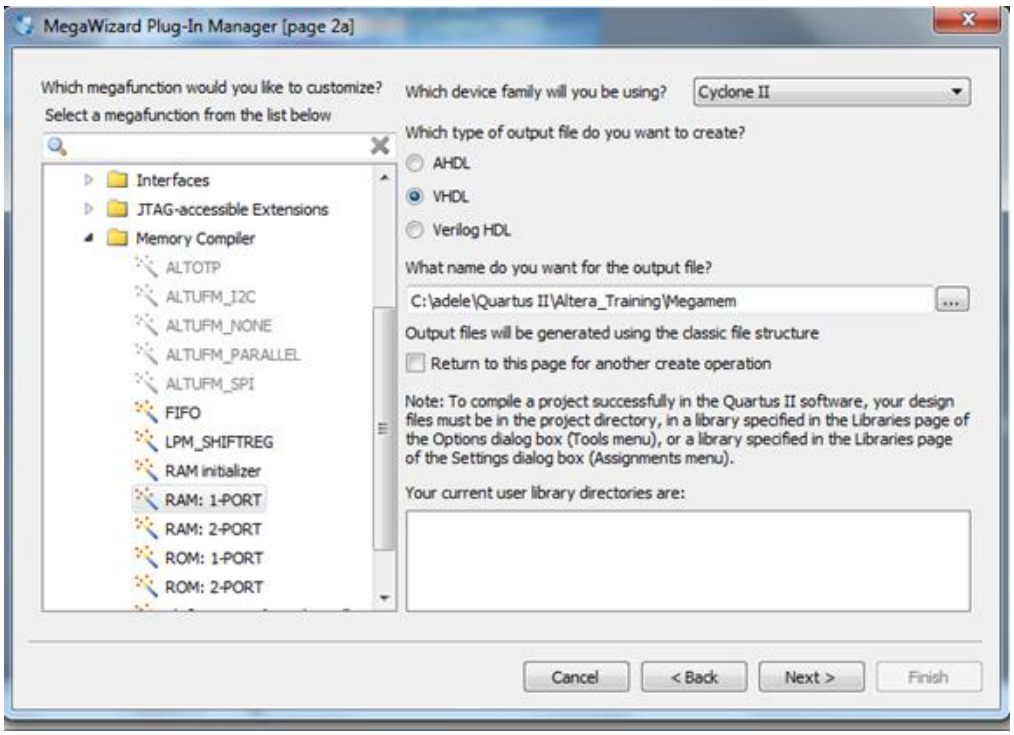


شکل (۴-۱) صفحه نخست استفاده از ابزار MegaWizard Plug-In Manager جهت استفاده از ابزارها و گیت‌های از پیش تعریف شده در Quartus.

در پنجره‌ی باز شده گزینه‌ی Create a New Custom Megafunction Variation را انتخاب می‌کنیم و Next را می‌زنیم.

در صفحه‌ی دوم در قسمت Installed Plug-Ins در لیست زیر مجموعه‌ی Memory Compiler گزینه‌ی RAM: 1-PORT را انتخاب می‌کنیم و دوباره Next را می‌زنیم. . در قسمت Which device family will you be using؟، Cyclone II را انتخاب می‌کنیم و نوع Output file را VHDL در نظر می‌گیریم و در نهایت نام Output file را انتخاب می‌کنیم که در اینجا نام آن را

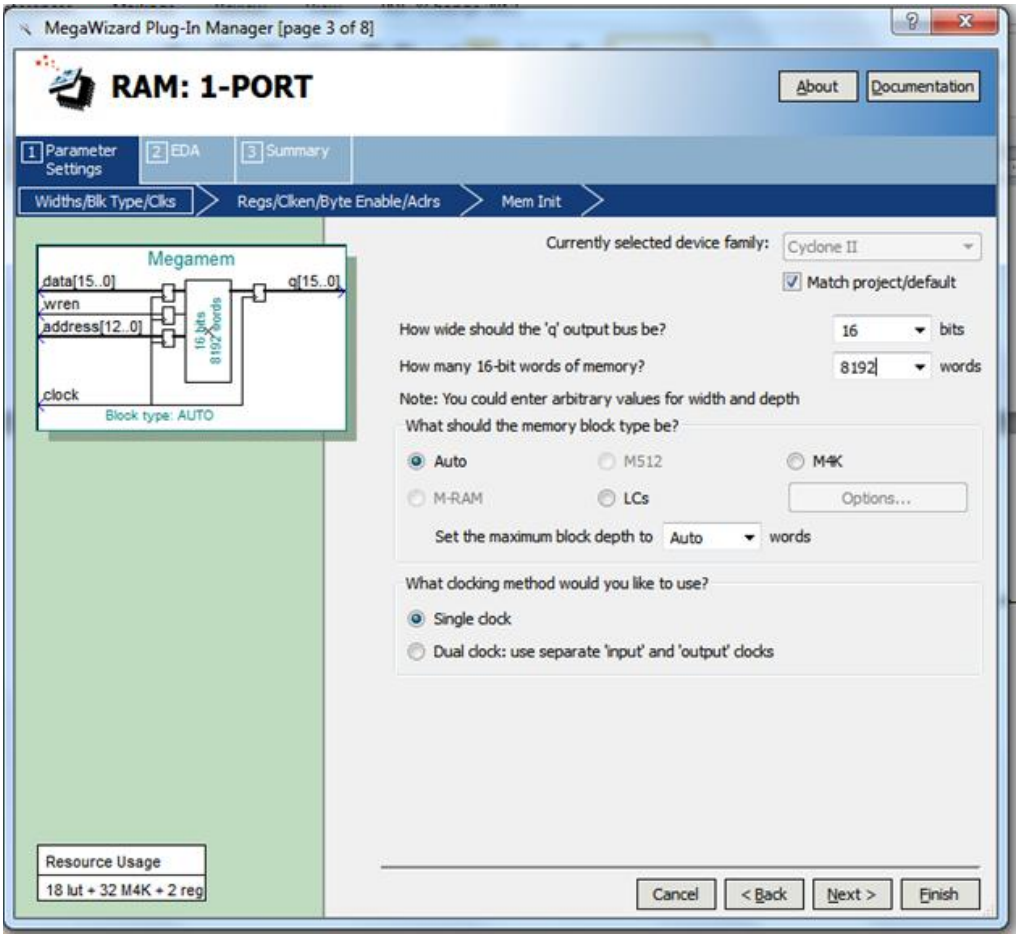
Megamem نامیده‌ایم. سپس دکمه‌ی Next را می‌زنیم.



شکل (۴-۲) نحوه استفاده از حافظه RAM موجود در Quartus و تعریف یک نمونه از آن با نام Megamem.

در صفحه‌ی ۳ تعداد خروجیهای q output bus را ۱۶ بیت در نظر گرفتیم و تعداد wordها را میتوان ۸۱۹۲ درنظر گرفت. تعداد wordهای لازم، بستگی به طول برنامه ی شما دارد. بنابراین با ۱۶ بیت خط آدرس می‌توان به ۶۴k خانه دسترسی پیدا کرد ولی ما به این مقدار زیاد از حافظه نیازی نداریم. حال کلید next را می‌زنیم تا به صفحه‌ی بعد برویم.

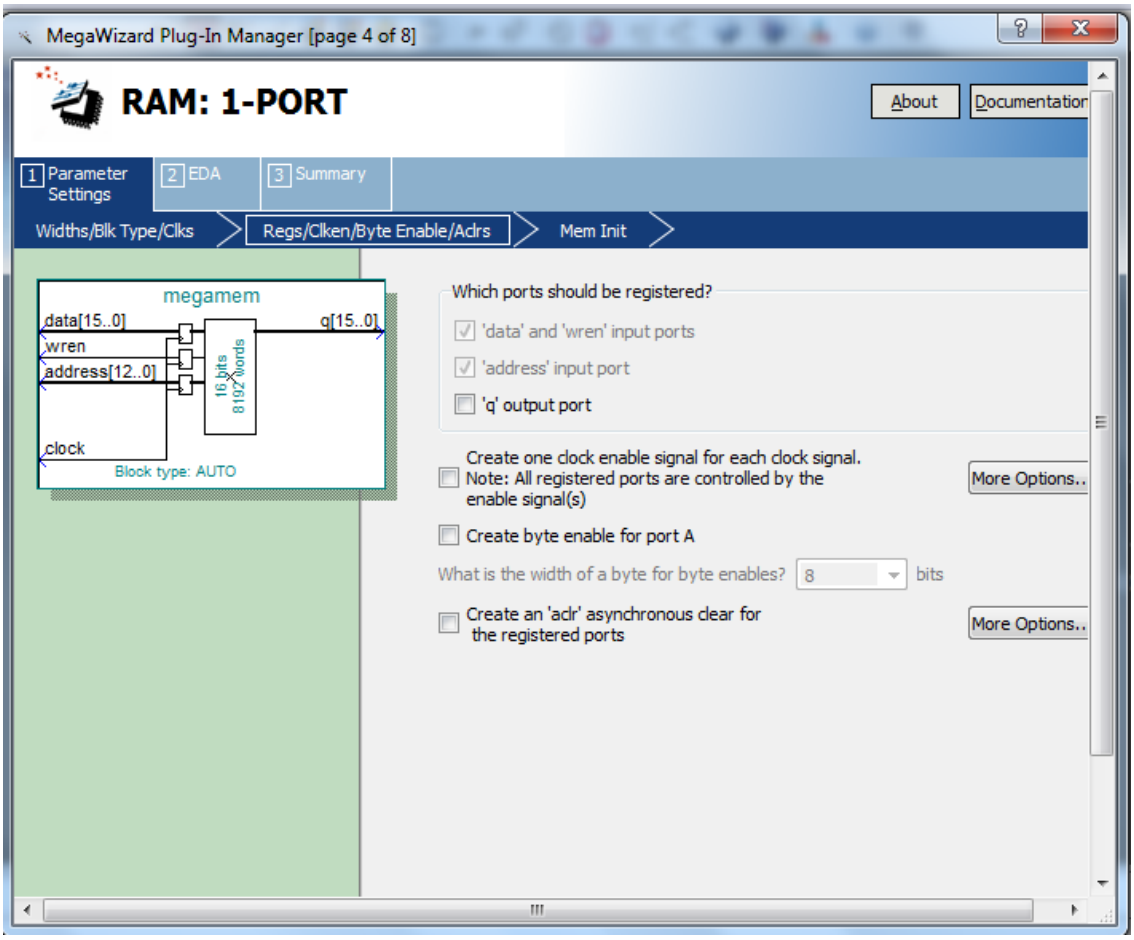
Copyright © 2015 ICEEP, University of Tehran



شکل (۴-۳) تعیین تعداد wordها و تعداد بیت‌های درون حافظه.

در صفحه‌ی چهارم تیک کنار 'q' را مانند شکل زیر بردارید و سپس کلید next را بزنید.

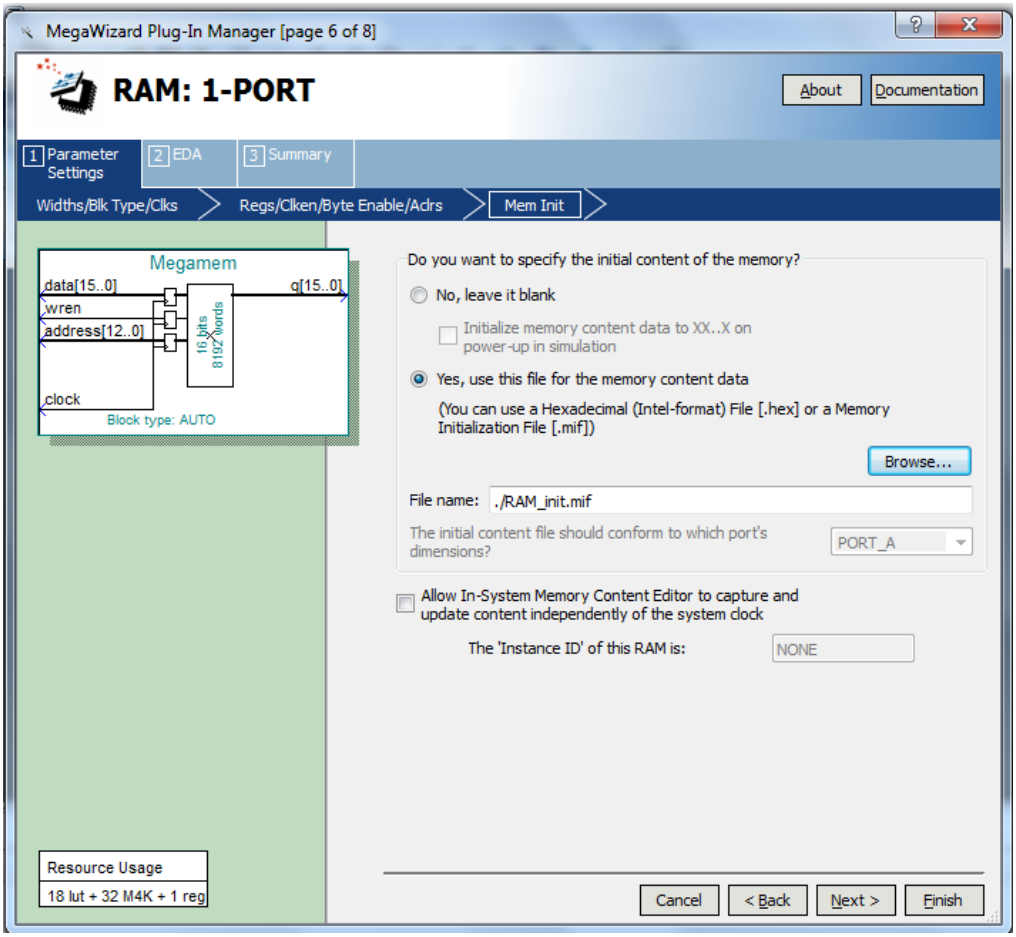
Copyright © 2015 ICEEP, University of Tehran



شکل (۴-۴) تنظیمات ریجیستر

برای قرار دادن دستور کارهای پروسسور در حافظه نیاز می‌باشد که ما یک سری مقادیر اولیه را در حافظه مقدار دهی کنیم. که این مقدار دهی اولیه یک بار انجام می‌شود و آن هم زمانی است که FPGA را Program می‌کنیم. این کار را می‌توان با استفاده از memory initialization file (MIF) نیز انجام داد. در اینجا ما یک فایل با نام RAM_init.mif داریم که در شاخه‌ای که درآن پروژه‌های Quartus II قرار دارد ذخیره کرده ایم. فورمت mif فایل خود را بر اساس آنچه در این فورمت گفته شده بنویسید. سپس روی "Yes, use this file for memory content data" کلیک کنید و فایل RAM_init.mif خود را مانند شکل ۵ اضافه کنید سپس دکمه ی Next را بزنید.

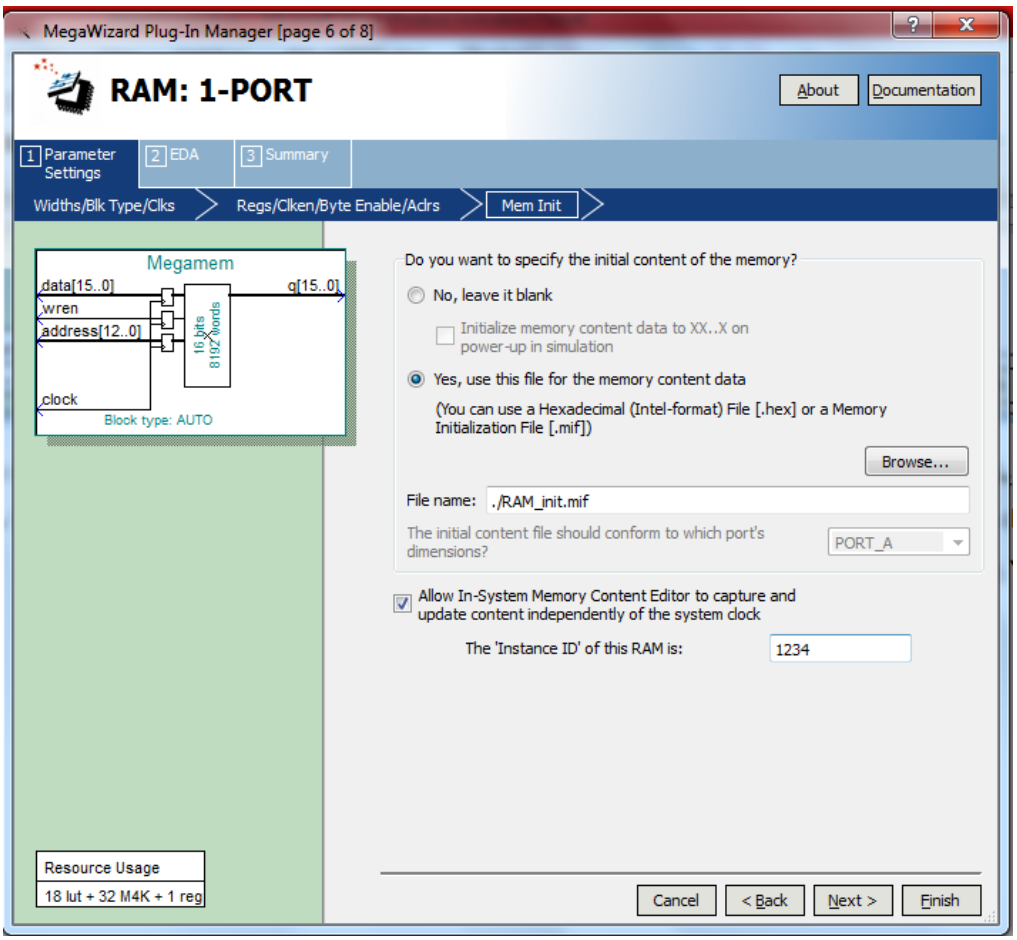
Copyright © 2015 ICEEP, University of Tehran



شکل (۴-۵) اضافه کردن فایل MIF

برای اینکه بتوانیم در زمان runtime تغییراتی در اطلاعات حافظه بدهیم باید " Allow in system memory content editor to capture and update content independently of the system clock "را مارک دار کنیم. 'instance ID' را می‌توان هر عدد ٤ رقمی‌مثل ١٢٣٤ در نظر گرفت.

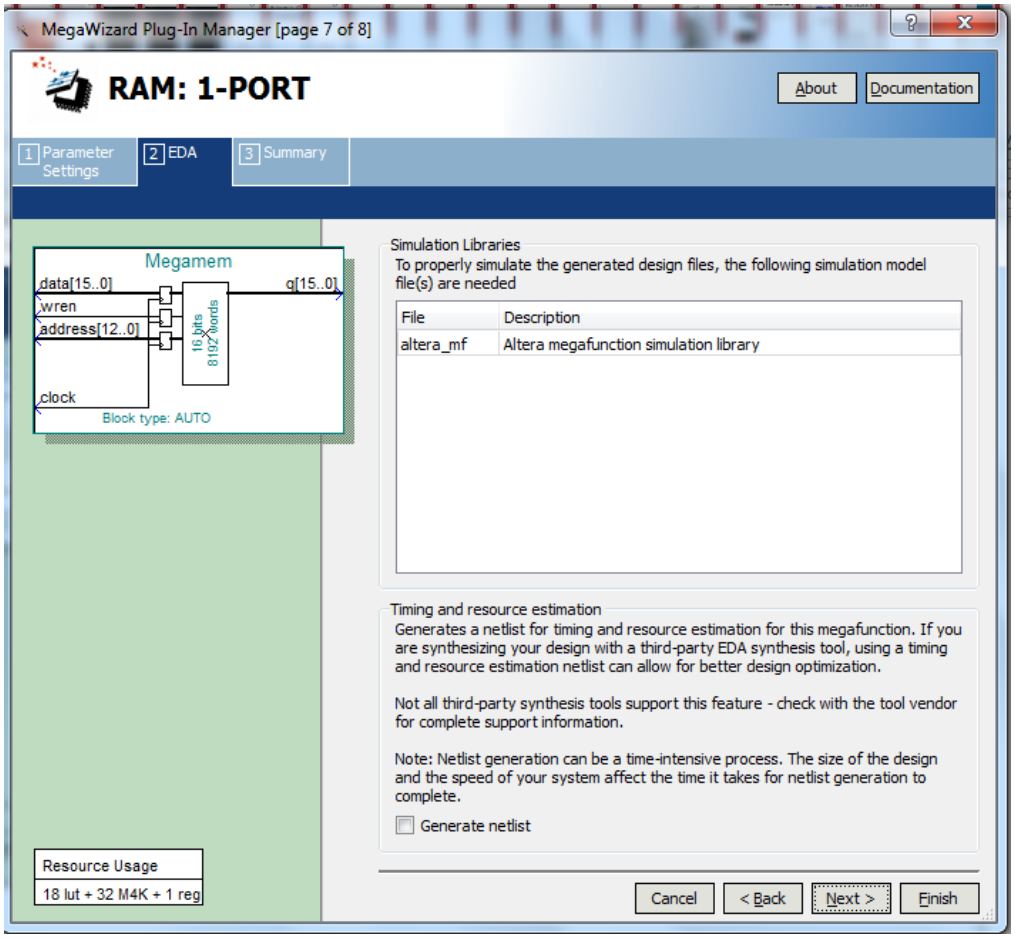
Copyright © 2015 ICEEP, University of Tehran



شکل (۴-۶) اجازه ی تغییر در زمان runtime

و در نهایت برای ماژول ram فایل .vhd در working directory ساخته میشود. روی next کلیک کنید تا به صفحه‌ی بعد وارد شوید.

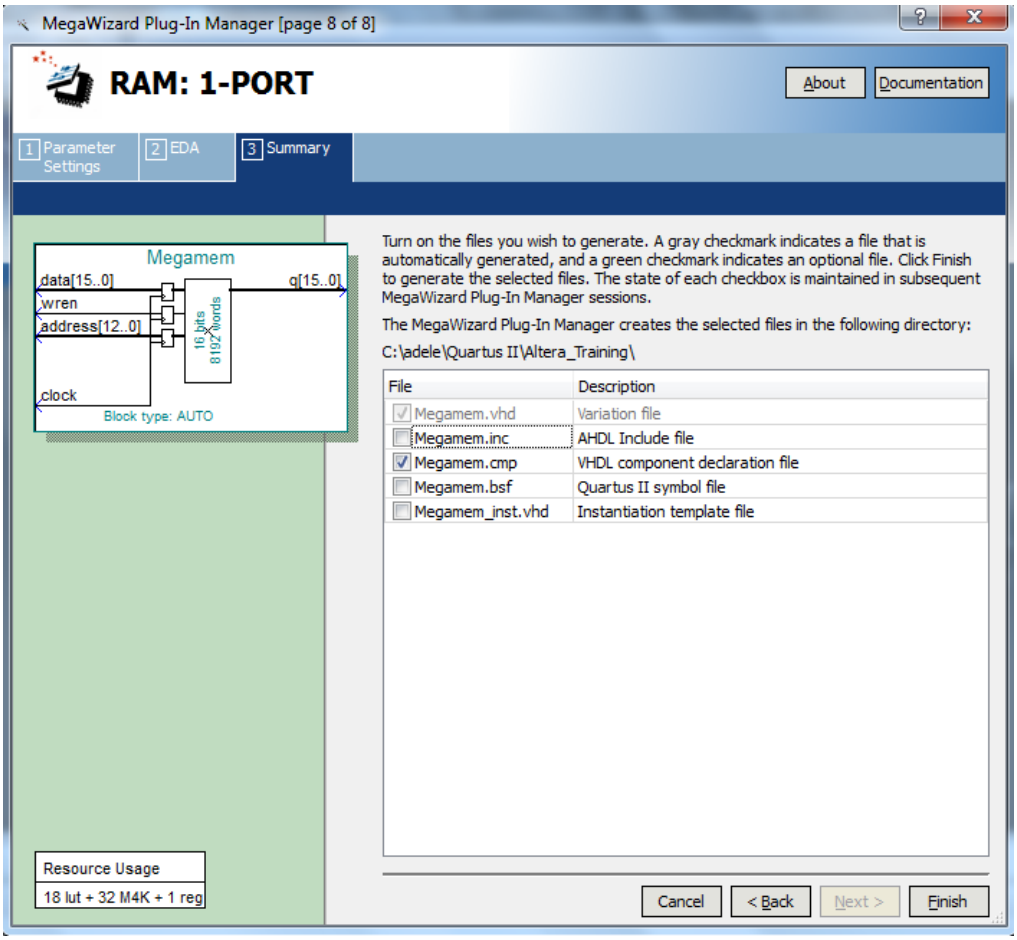
Copyright © 2015 ICEEP, University of Tehran



شکل (۴-۷) کتابخانه‌های شبیه سازی سشده

در نهایت در قسمت summary، خلاصه‌ی تمام فایل‌های ساخته شده را می‌توان مشاهده کرد.

Copyright © 2015 ICEEP, University of Tehran



شکل (۴-۸) خلاصه ی تمام فایل‌های ساخته شده

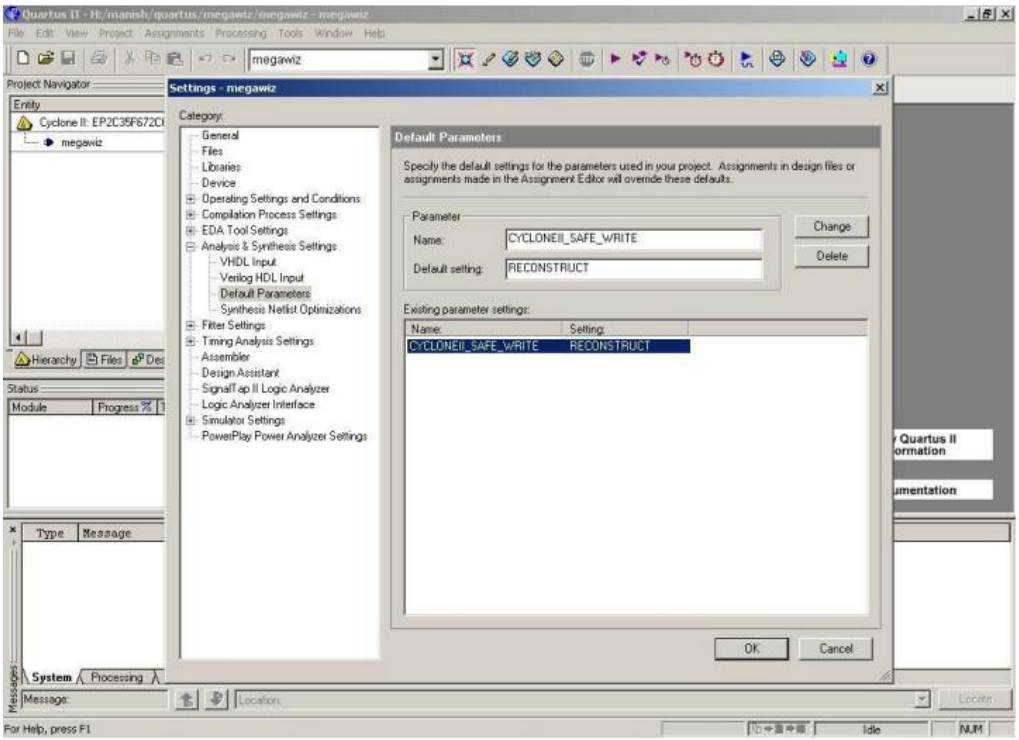
Copyright © 2015 ICEEP, University of Tehran

فصل پنجم

نحوه استفاده از ابزار

In-System Memory Content Editor

قبل از اینکه شما بتوانید از ابزار In-System Memory Content Editor استفاده کنید یک سری تنظیمات دیگر لازم به اجرا می‌باشد. برای این کار در Quartus II از منوی Assignment گزینه‌ی Setting را انتخاب کنید تا پنجره‌ای مانند شکل زیر باز شود. در قسمت Category عنوان Analysis and Synthesis را انتخاب کنید و در زیر منوی باز شده عبارت Default Parameter را برگزینید. همانطور که در شکل مشاهده می‌کنید در قسمت parameter name عبارت CYCLONEII_SAFE_WRITE را تایپ کرده‌ایم و در قسمت Default setting عبارت RECONSTRUCT را تایپ کرده‌ایم. این پارمترها به ابزار سنتز Quartus II اجازه می‌دهند تا در صورت لزوم به single-port RAM اجازه تغییر برای انجام عملیات reading و writing در حافظه داده شود. روی ok کلیک کنید.



شکل (۵-۱) تنظیمات اولیه

Copyright © 2015 ICEEP, University of Tehran

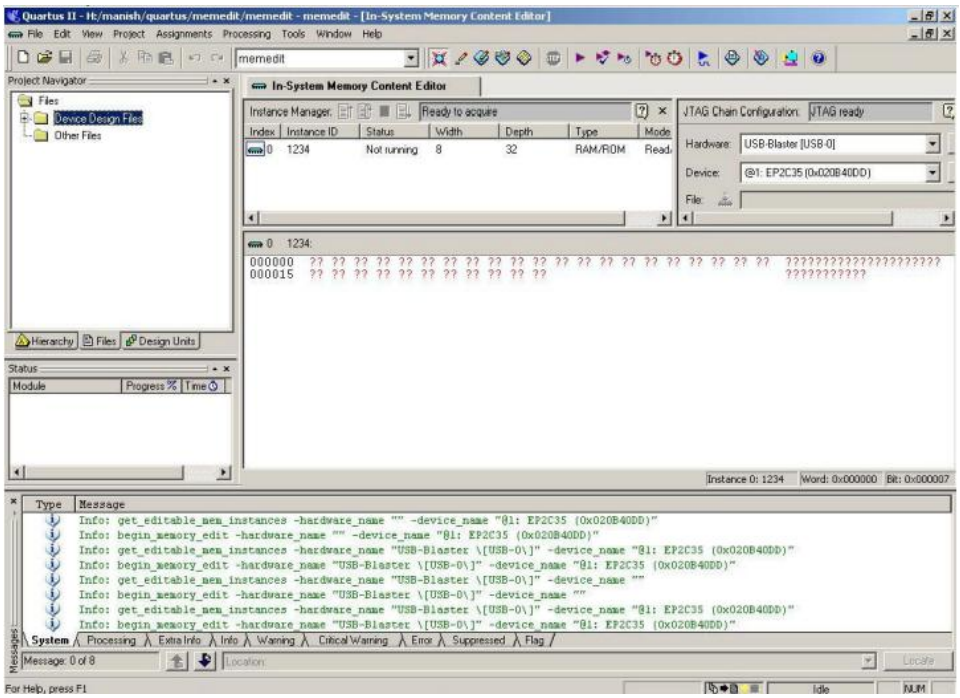
حال می‌توانید از این حافظه به وسیله‌ی port mapping در طراحی خودتان استفاده کنید.

اکنون کد را کامپایل نمایید و مدار را در برد DE2 دانلود کنید. بعد از دانلود طراحی در FPGA ،زیر منوی Tools > In system Memory Content Editor را مانند شکل زیر انتخاب کنید. برای اطمینان از اتصال برد DE2 بر روی دکمه‌ی Setup در قسمت راست صفحه کلید کنید. در شکل زیر USB_Blaster hardware را انتخاب کنید سپس hardware setup dialog را ببندید.

نکته: در هنگام انجام این مرحله ممکن است برد FPGA روشن شود ودر حالت RUN قرار بگیرد.

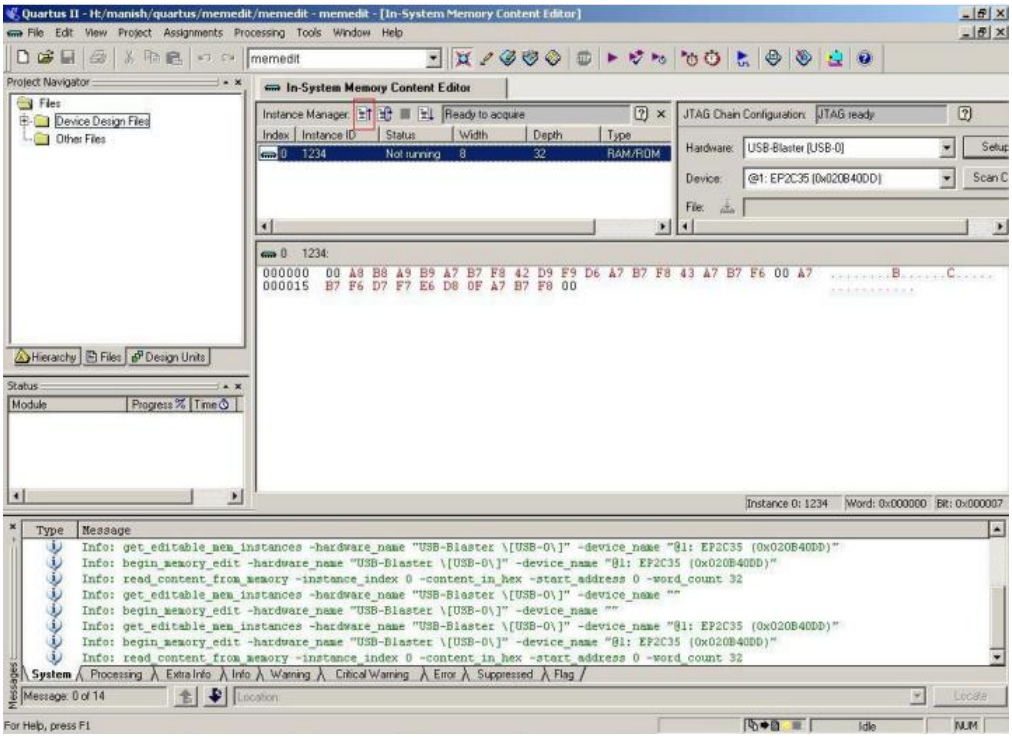
شما می‌توانید مشاهده کنید که device ID مشخصی در قسمت instance manager ظاهر می‌شود.

شرایط برای دریافت آماده است و ما با دیدن "؟؟" می‌توانیم متوجه شویم که حافظه هنوز آماده نیست



شکل (۵-۲) شمای کلی In system Memory Content Editor

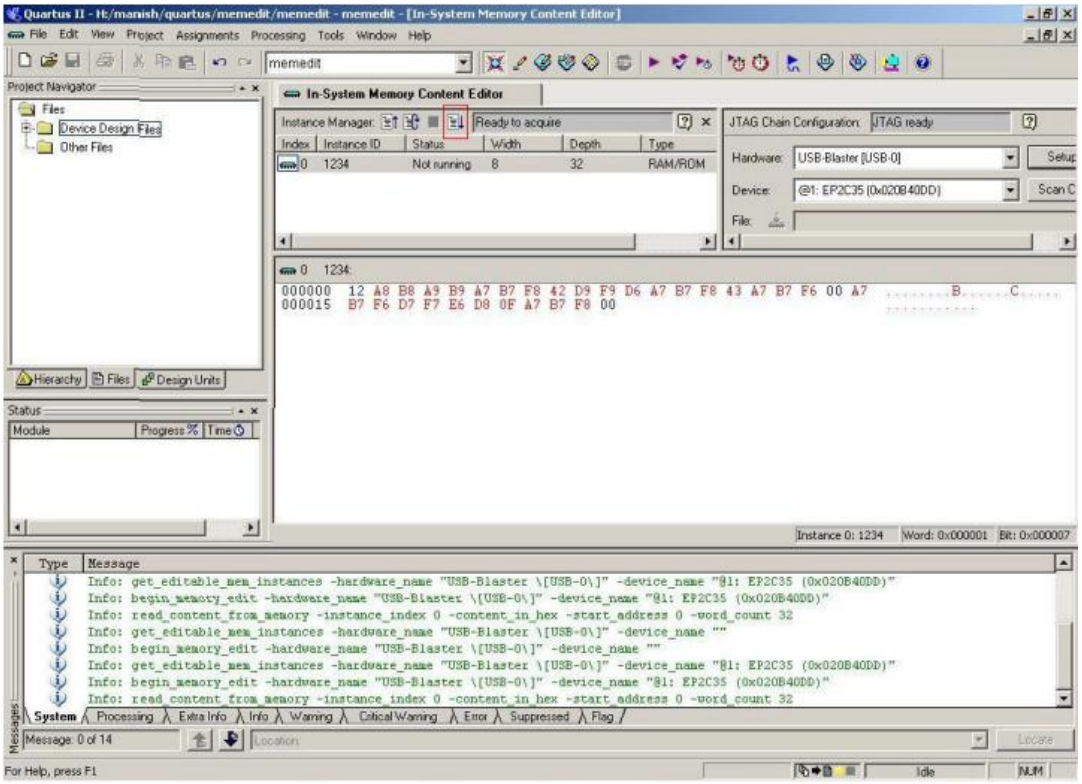
Memory را در قسمت Instance manager انتخاب کنید وهمان طور که در شکل زیر می‌بینید بر روی دکمه‌ی read data from system memory کلیک کنید، حال شما می‌توانید داخل حافظه را ببینید.



شکل (۵-۳) نمایش اطلاعات داخل حافظه

می‌توان با نوشتن مستقیم بر روی خانه‌های حافظه محتوای درون آنها را تغییر داد. برای مثال در شکل زیر محتوای خانه‌ی حافظه‌ی 000000 از 00 به 12 تغییر کرده است. این تنها یک تغییر است که ما در ویرایشگر اعمال نموده ایم و برای اینکه این عملیات نوشتن به صورت واقعی در حافظه نیز انجام شود باید بر روی دکمه‌ی write که در کادر قرمز رنگ مشاهده می‌شود کلیک کنید.

Copyright © 2015 ICEEP, University of Tehran



شکل (۵-۴) تغییر در حافظه

بعد از انجام تغییرات بر روی محتوای خانه‌های حافظه لزومی‌به recompile کردن کل طراحی در FPGA نمی‌باشد. کافیست مدار خود را ریست کرده تا طراحی شما این بار با مقادیر جدید حافظه به کار خود ادامه دهد. همچنین این run time memory content editor می‌تواند برای تغییر در یک یا دو operand در برنامه ی شما نیز استفاده شود.

Copyright © 2015 ICEEP, University of Tehran