EUCLID

INTEL 8051 MICROCONTROLLER PROJECT

Embedded Systems Project

Agnese Salutari, agnese.salutari@student.univaq.it

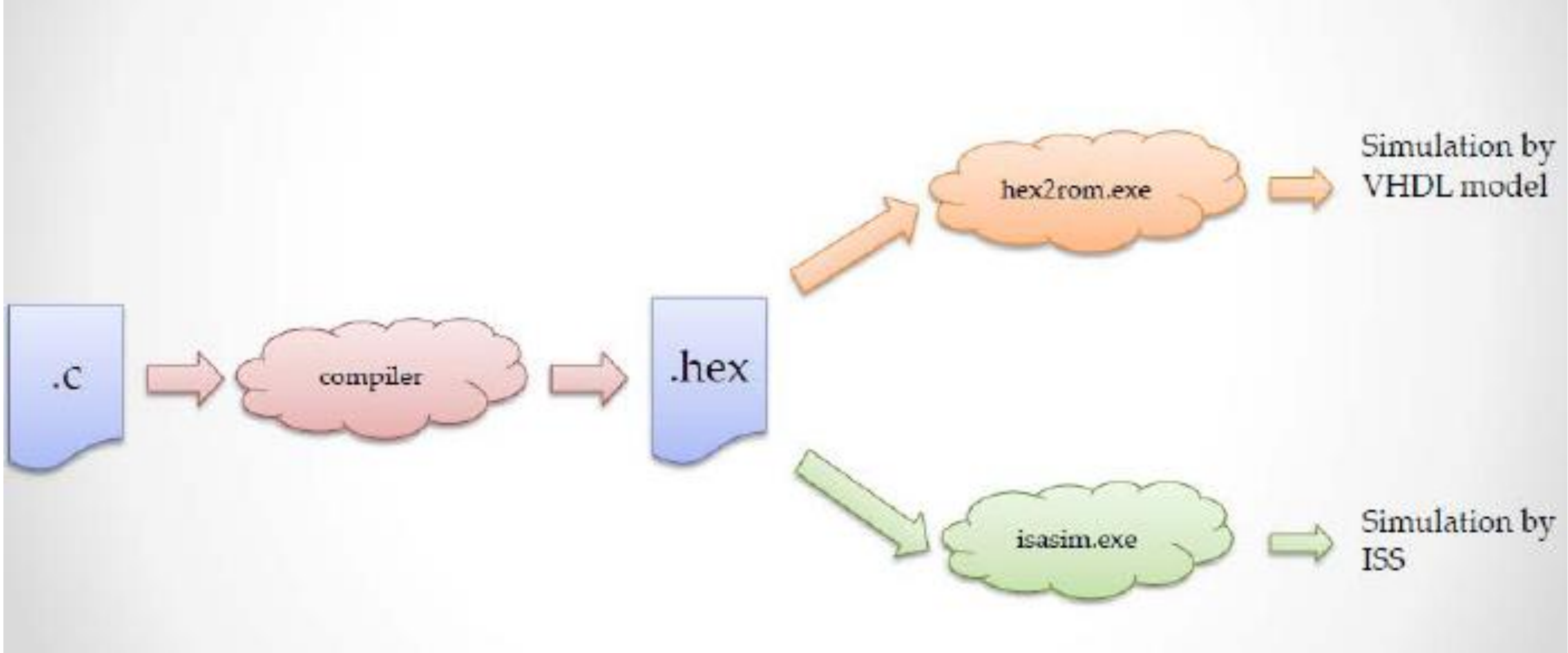
EUCLID PROJECT OVERWIEW

Euclid Project consists of a C program tested via ISS and VHDL.

This program performs Euclid's Algorithm to find the value of GCD(x, y), that is the Greatest Common Divisor of two integers:

- $a_i = b_i * q_i + r_i$; $i = 0, 1, 2, \dots, n$ is the step number
- $a_0 = x$; $b_0 = y$
- $q_i = a_i / b_i$; $r_i = a_i \% b_i$
- $a_{i+1} = b_i$; $b_{i+1} = r_i$
- $a_n = b_n * q_n + 0$; *the last step is reached when r is 0;* GCD(x, y) = b_n

EUCLID PROJECT OVERWIEW



ENVIRONMENT

In a first time, I tried to develop this project on Kubuntu 18, but Vivado wasn't compatible with it, so I finally developed it on Windows 10 Pro.

I used the following tools:

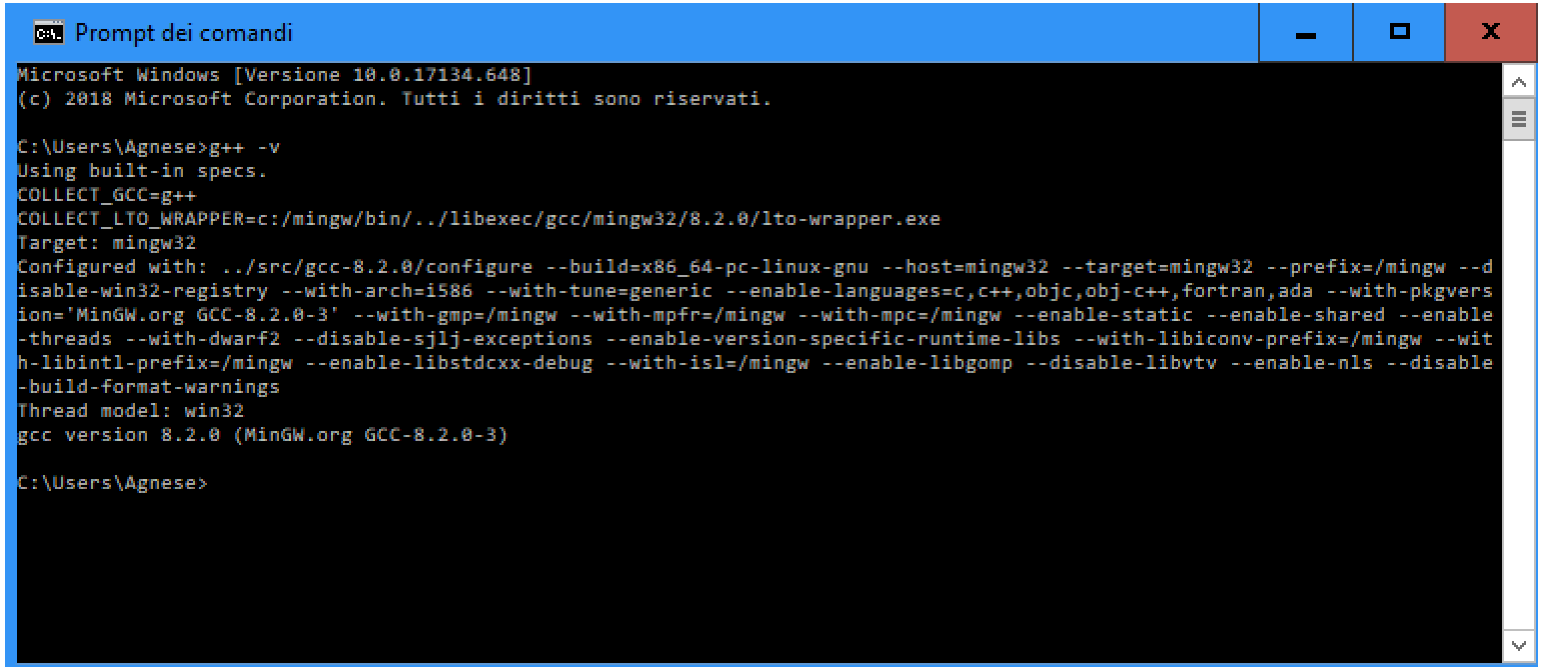
- C compilers:
  - MingGW (needed on Windows): http://www.mingw.org/
  - SDCC: http://sdcc.sourceforge.net/
  - Keil C51: http://www.keil.com/
- For ISS simulations:
  - Dalton Project ISS and ISASim: http://www.ann.ece.ufl.edu/i8051/
- For VHDL simulations:
  - Dalton Project VHDL and Hex2Rom: http://www.ann.ece.ufl.edu/i8051/
  - Vivado Web Package: https://www.xilinx.com/products/design-tools/vivado.html

GETTING READY... G++

MingGW allows you to run g++ commands on Windows.

After downloading and installation, don't forget to add MingGW path to environment variable path!

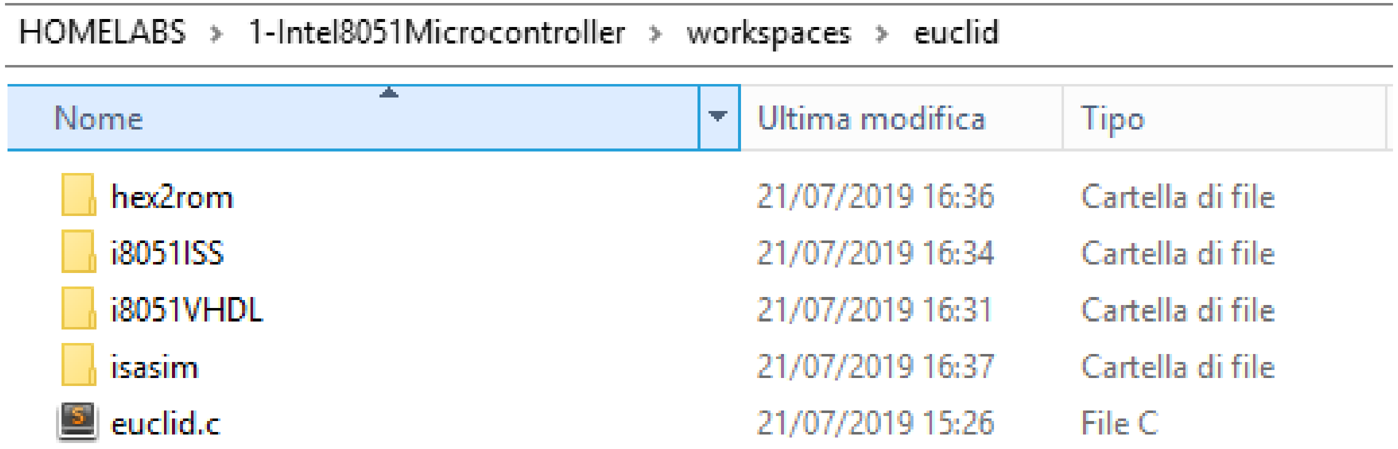
If the installation was successfull, by typing g++ -v in Command Prompt, you should see something like the following:



```
Microsoft Windows [Versione 10.0.17134.648]
(c) 2018 Microsoft Corporation. Tutti i diritti sono riservati.

C:\Users\Agnese>g++ -v
Using built-in specs.
COLLECT_GCC=g++
COLLECT_LTO_WRAPPER=c:/mingw/bin/../libexec/gcc/mingw32/8.2.0/lto-wrapper.exe
Target: mingw32
Configured with: ../src/gcc-8.2.0/configure --build=x86_64-pc-linux-gnu --host=mingw32 --target=mingw32 --prefix=/mingw --d
isable-win32-registry --with-arch=i586 --with-tune=generic --enable-languages=c,c++,objc,obj-c++,fortran,ada --with-pkgvers
ion='MinGW.org GCC-8.2.0-3' --with-gmp=/mingw --with-mpfr=/mingw --with-mpc=/mingw --enable-static --enable-shared --enable
-threads --with-dwarf2 --disable-sjlj-exceptions --enable-version-specific-runtime-libs --with-libiconv-prefix=/mingw --wit
h-libintl-prefix=/mingw --enable-libstdcxx-debug --with-isl=/mingw --enable-libgomp --disable-libvtv --enable-nls --disable
-build-format-warnings
Thread model: win32
gcc version 8.2.0 (MinGW.org GCC-8.2.0-3)

C:\Users\Agnese>
```

GETTING READY…WORKSPACE

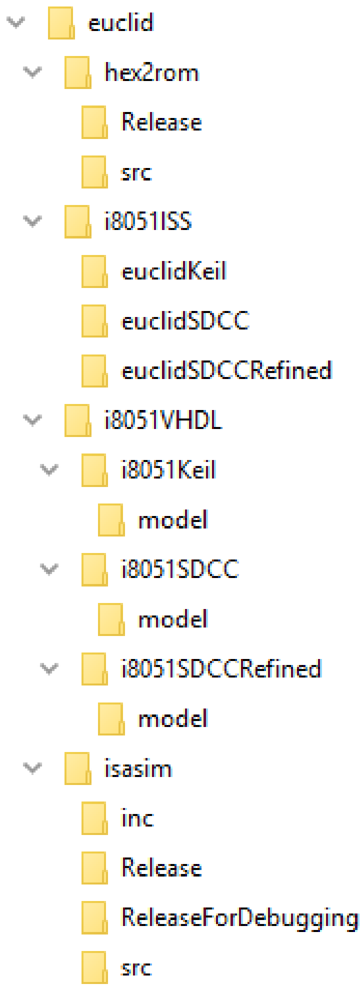
The Project structure is the following:



HOMELABS › 1-Intel8051Microcontroller › workspaces › euclid

Nome	Ultima modifica	Tipo
hex2rom	21/07/2019 16:36	Cartella di file
i8051ISS	21/07/2019 16:34	Cartella di file
i8051VHDL	21/07/2019 16:31	Cartella di file
isasim	21/07/2019 16:37	Cartella di file
euclid.c	21/07/2019 15:26	File C

- i8051ISS contains ISS simulation files
- i8051VHDL folder contains VHDL simulation files



```
euclid
  hex2rom
    Release
    src
  i8051ISS
    euclidKeil
    euclidSDCC
    euclidSDCCRefined
  i8051VHDL
    i8051Keil
      model
    i8051SDCC
      model
    i8051SDCCRefined
      model
  isasim
    inc
    Release
    ReleaseForDebugging
    src
```

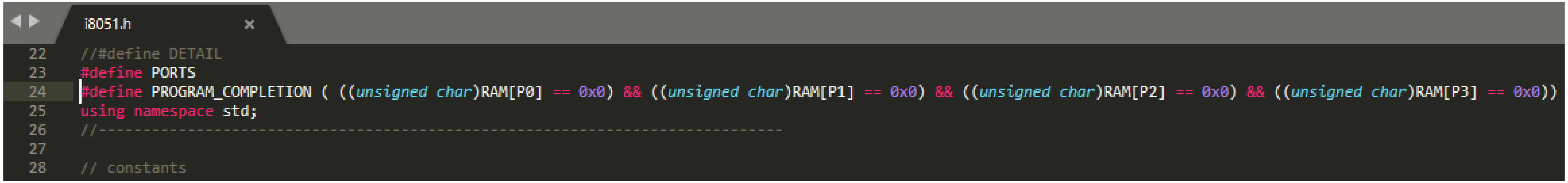
GETTING READY…WORKSPACE

From http://www.ann.ece.ufl.edu/i8051/i8051syn/ I downloaded:

- i8051.cc and main.cc files into euclid\isasim\src
- i8051.h file into euclid\isasim\inc
- i8051_mkr.c file into euclid\hex2rom\src
- i8051_all.vhd, i8051_alu.vhd, i8051_ctr.vhd, i8051_dbg.vhd, i8051_dec.vhd, i8051_lib.vhd, i8051_ram.vhd, i8051_rom.vhd, i8051_tsb.vhd and i8051_xrm.vhd files into euclid\i8051VHDL\i8051SDCC, euclid\i8051VHDL\i8051SDCCRefined\model and euclid\i8051VHDL\i8051Keil\model

UPDATING THE CODE

- In euclid\isasim\inc\i8051.h file, I modified line 24 and added line 25, as you can see below:

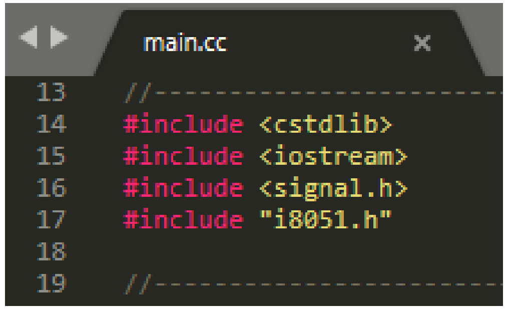


i8051.h

```
22  //#define DETAIL
23  #define PORTS
24  #define PROGRAM_COMPLETION ( ((unsigned char)RAM[P0] == 0x0) && ((unsigned char)RAM[P1] == 0x0) && ((unsigned char)RAM[P2] == 0x0) && ((unsigned char)RAM[P3] == 0x0))
25  using namespace std;
26  //-----------------------------------------------------------------------------
27  
28  // constants
```

Line 24 describes the program completion conditions: the program ends when all those conditions are true.

- In file euclid\isasim\src\main.cc, I updated dependencies like that:



main.cc

```
//----------------------
#include <cstdlib>
#include <iostream>
#include <signal.h>
#include "i8051.h"

//----------------------
```

BUILDING ISASIM RELEASE

On Command Prompt, from euclid\isasim:

- To compile and assemble i80151.cc file:

```
g++ -I "./inc" -O3 -Wall -c -o "./Release/i8051.o" "./src/i8051.cc"
```

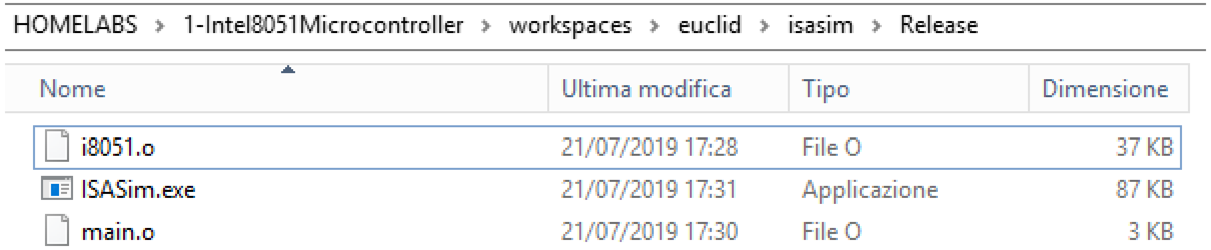
- To compile and assemble main.cc file:

```
g++ -I "./inc" -O3 -Wall -c -o "./Release/main.o" "./src/main.cc"
```

- To link previously obtained object files:

```
g++ -I "./inc" -o "./Release/ISASim" ./Release/*.o
```

Finally, there are 3 new files in euclid\isasim\Release folder:



HOMELABS › 1-Intel8051Microcontroller › workspaces › euclid › isasim › Release

Nome	Ultima modifica	Tipo	Dimensione
i8051.o	21/07/2019 17:28	File O	37 KB
ISASim.exe	21/07/2019 17:31	Applicazione	87 KB
main.o	21/07/2019 17:30	File O	3 KB

BUILDING ISASIM RELEASE FOR DEBUGGING

On Command Prompt, from euclid\isasim:

- To compile and assemble i80151.cc file:

```
g++ -I "./inc" -O3 -Wall -c -o "./ReleaseForDebugging/i8051.o" -DDEBUG -DDEBUG_PC -DDETAIL "./src/i8051.cc"
```

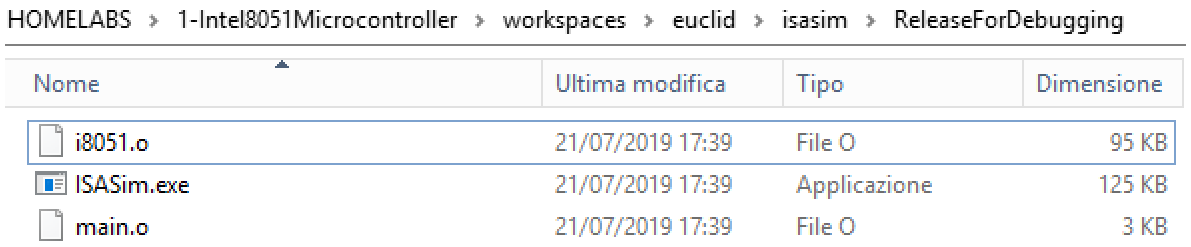
- To compile and assemble main.cc file:

```
g++ -I "./inc" -O3 -Wall -c -o "./ReleaseForDebugging/main.o" -DDEBUG -DDEBUG_PC -DDETAIL "./src/main.cc"
```

- To link previously obtained object files:

```
g++ -I "./inc" -o "./ReleaseForDebugging/ISASim" -DDEBUG -DDEBUG_PC -DDETAIL ./ReleaseForDebugging/i8051.o
./ReleaseForDebugging/main.o
```

Finally, there are 3 new files in euclid\isasim\ReleaseForDebugging folder:



HOMELABS › 1-Intel8051Microcontroller › workspaces › euclid › isasim › ReleaseForDebugging

Nome	Ultima modifica	Tipo	Dimensione
i8051.o	21/07/2019 17:39	File O	95 KB
ISASim.exe	21/07/2019 17:39	Applicazione	125 KB
main.o	21/07/2019 17:39	File O	3 KB

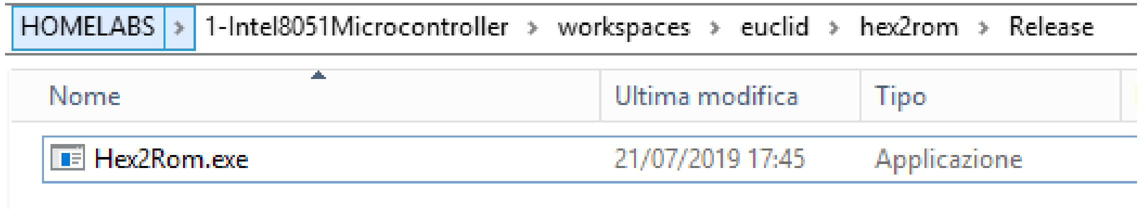
BUILDING HEX2ROM

On Command Prompt, from euclid\hex2rom:

- To compile and assemble i8051_mkr.c file:

```
gcc -O3 -Wall -o "./Release/Hex2Rom" "./src/i8051_mkr.c"
```

Finally, there should be a new file in euclid\hex2rom\Release folder:

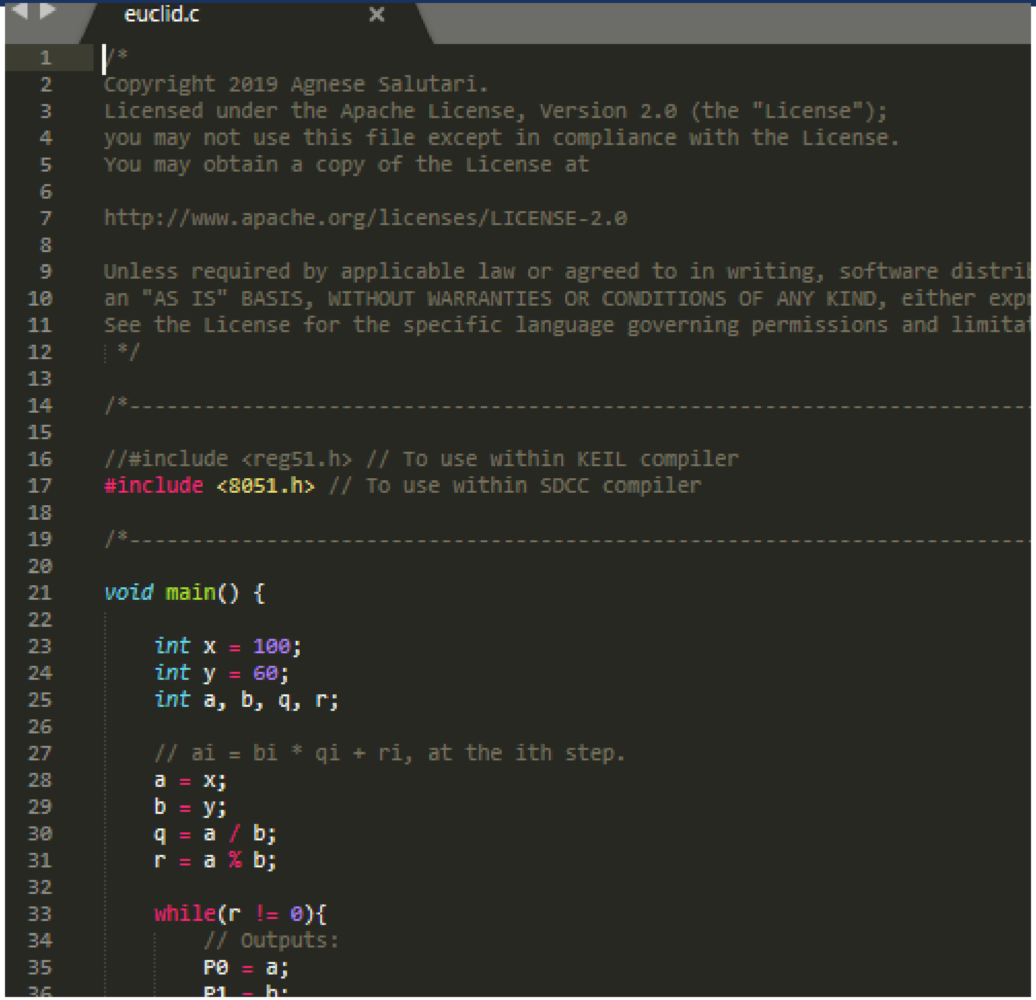


HOMELABS › 1-Intel8051Microcontroller › workspaces › euclid › hex2rom › Release

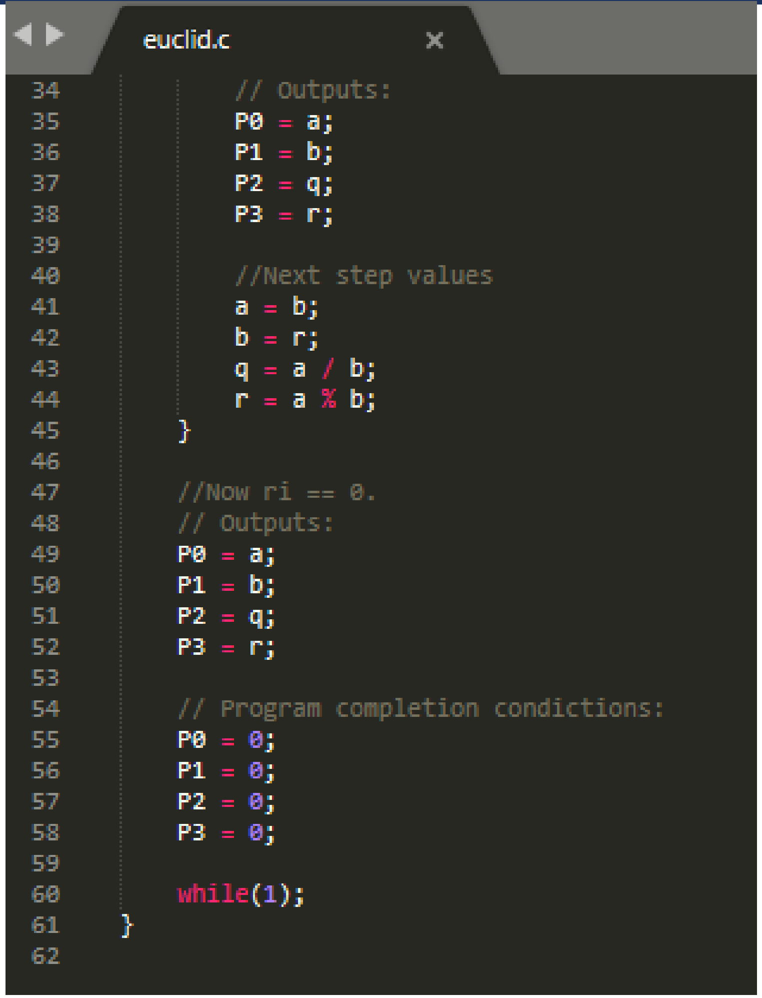
Nome	Ultima modifica	Tipo
Hex2Rom.exe	21/07/2019 17:45	Applicazione

This time we didn't use -c inside the command, (-c = compile and assemble but do not link the input file) so there is only one output file because gcc command compiled and linked the input file at once.

C PROGRAM: EUCLID.C



```
/*
Copyright 2019 Agnese Salutari.
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distri
an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either expr
See the License for the specific language governing permissions and limita
 */

/*--------------------------------------------------------------------------

//#include <reg51.h> // To use within KEIL compiler
#include <8051.h> // To use within SDCC compiler

/*--------------------------------------------------------------------------

void main() {

    int x = 100;
    int y = 60;
    int a, b, q, r;

    // ai = bi * qi + ri, at the ith step.
    a = x;
    b = y;
    q = a / b;
    r = a % b;

    while(r != 0){
        // Outputs:
        P0 = a;
        P1 = b;
```



```
        // Outputs:
        P0 = a;
        P1 = b;
        P2 = q;
        P3 = r;

        //Next step values
        a = b;
        b = r;
        q = a / b;
        r = a % b;
    }

    //Now ri == 0.
    // Outputs:
    P0 = a;
    P1 = b;
    P2 = q;
    P3 = r;

    // Program completion condictions:
    P0 = 0;
    P1 = 0;
    P2 = 0;
    P3 = 0;

    while(1);
}
```

ALGORITHM FLOW

In euclid.c file: x = 100; y=60;

So the program follows this flow:

- $a_i = \mathrm{b}_i * \mathrm{q}_i + \mathrm{r}_i$; $\mathrm{i} = 0, 1, 2, \dots, n$ is the step number
- $100 = 60 * 1 + 40;$
- $60 = 40 * 1 + 20;$
- $40 = 20 * 2 + 0;$

Then GCD(x, y) = 20.

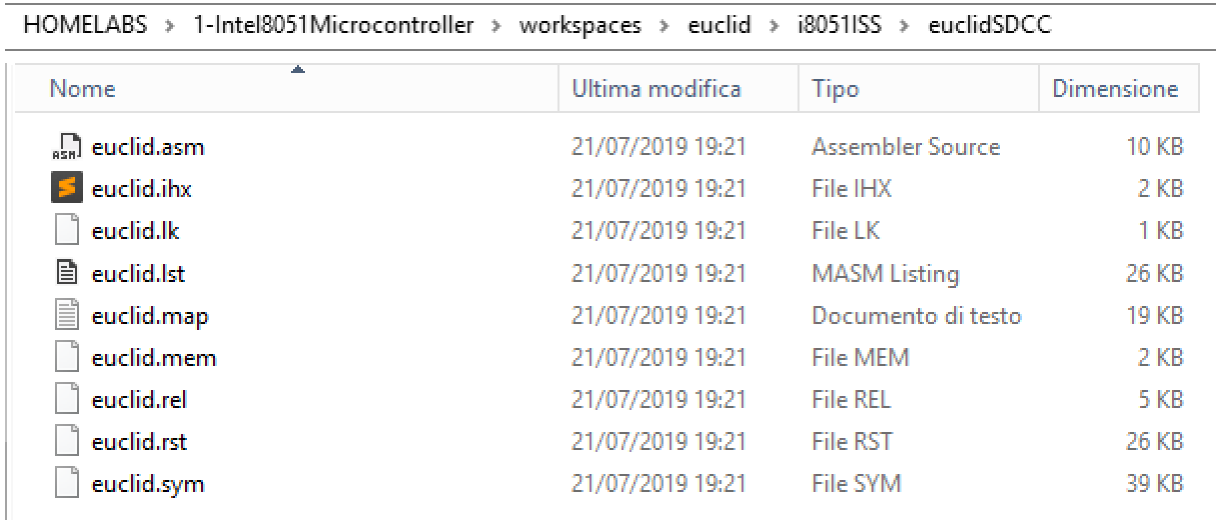
Decimal to Hexadecimal: 100 = 0x64; 60 = 0x3C; 40 = 0x28; 20 = 0x14; 2 = 0x02; 0 = 0x00.

SDCC BUILDING EUCLID.C

On Command Prompt, from euclid\i8051ISS\euclidSDCC:

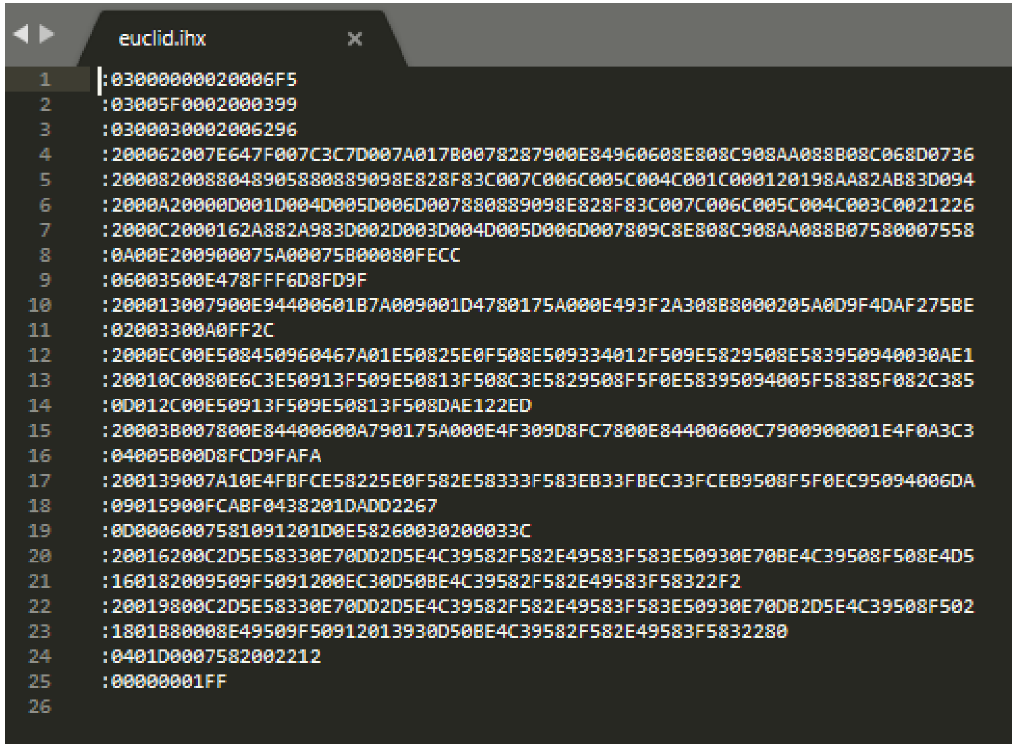
sdcc ../../euclid.c

Finally, in euclid\i8051ISS\euclidSDCC folder there should be the following files:



HOMELABS › 1-Intel8051Microcontroller › workspaces › euclid › i8051ISS › euclidSDCC

Nome	Ultima modifica	Tipo	Dimensione
euclid.asm	21/07/2019 19:21	Assembler Source	10 KB
euclid.ihx	21/07/2019 19:21	File IHX	2 KB
euclid.lk	21/07/2019 19:21	File LK	1 KB
euclid.lst	21/07/2019 19:21	MASM Listing	26 KB
euclid.map	21/07/2019 19:21	Documento di testo	19 KB
euclid.mem	21/07/2019 19:21	File MEM	2 KB
euclid.rel	21/07/2019 19:21	File REL	5 KB
euclid.rst	21/07/2019 19:21	File RST	26 KB
euclid.sym	21/07/2019 19:21	File SYM	39 KB



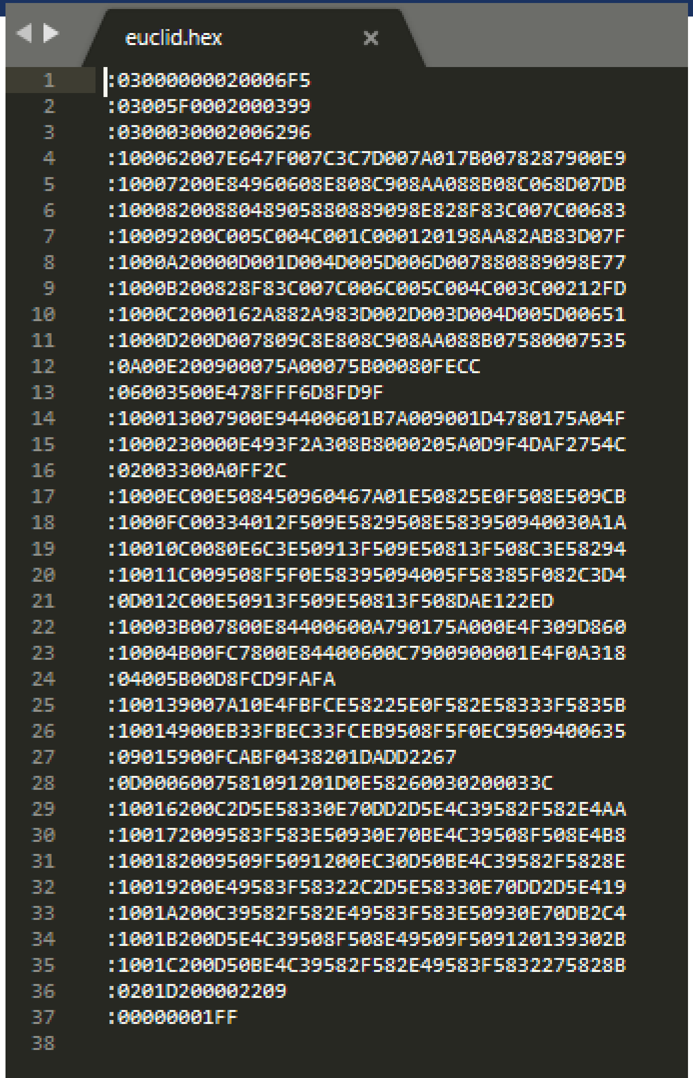
```
:03000000020006F5
:03005F0002000399
:0300030002006296
:200062007E647F007C3C7D007A017B0078287900E84960608E808C908AA088B08C068D0736
:2000820088048905880889098E828F83C007C006C005C004C001C000120198AA82AB83D094
:2000A20000D001D004D005D006D00788088909 8E828F83C007C006C005C004C003C0021226
:2000C2000162A882A983D002D003D004D005D006D007809C8E808C908AA088B07580007558
:0A00E200900075A00075B00080FECC
:06003500E478FFF6D8FD9F
:200013007900E94400601B7A009001D4780175A000E493F2A308B8000205A0D9F4DAF275BE
:02003300A0FF2C
:2000EC00E508450960467A01E50825E0F508E509334012F509E5829508E58395094003 0AE1
:20010C0080E6C3E50913F509E50813F508C3E5829508F5F0E58395094005F58385F082C385
:0D012C00E50913F509E50813F508DAE122ED
:20003B007800E84400600A790175A000E4F309D8FC7800E84400600C7900900001E4F0A3C3
:04005B00D8FCD9FAFA
:200139007A10E4FBFCE58225E0F582E58333F583EB33FBEC33FCEB9508F5F0EC95094006DA
:09015900FCABF0438201DADD2267
:0D00060075810912 01D0E58260030200033C
:20016200C2D5E58330E70DD2D5E4C39582F582E49583F583E50930E70BE4C39508F508E4D5
:160182009509F5091200EC30D50BE4C39582F582E49583F58322F2
:20019800C2D5E58330E70DD2D5E4C39582F582E49583F583E50930E70DB2D5E4C39508F502
:1801B80008E49509F50912013930D50BE4C39582F582E49583F5832280
:0401D00075820022 12
:00000001FF

```

SDCC BUILDING EUCLID.C

On Command Prompt, from euclid\i8051ISS\euclidSDCC:

packihx euclid.ihx **>** euclid.hex

Finally, euclid.hex file should have been created from euclid.ihx.



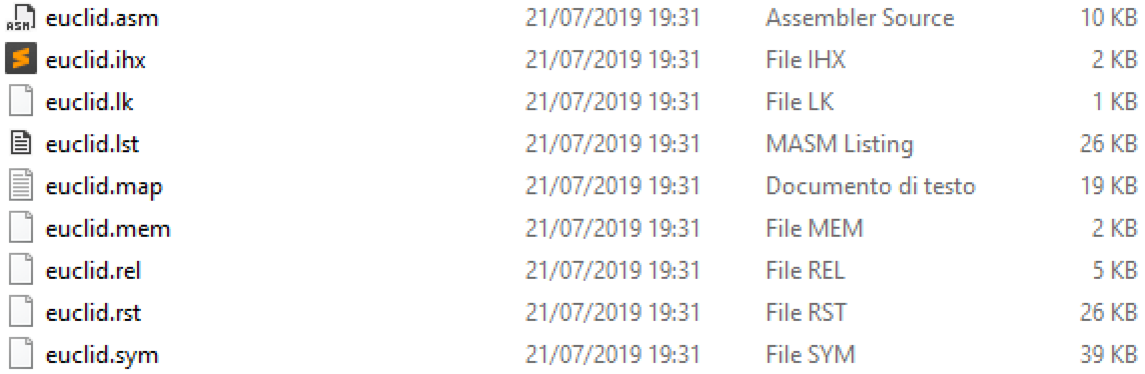
```
:03000000020006F5
:03005F0002000399
:0300030002006296
:100062007E647F007C3C7D007A017B0078287900E9
:10007200E84960608E808C908AA088B08C068D07DB
:100082008880489058808890898E828F83C007C00683
:10009200C005C004C001C000120198AA82AB83D07F
:1000A20000D001D004D005D006D00788088898E77
:1000B200828F83C007C006C005C004C003C00212FD
:1000C2000162A882A983D002D003D004D005D00651
:1000D200D007809C8E808C908AA088B07580007535
:0A00E200900075A00075B00080FECC
:06003500E478FFF6D8FD9F
:100013007900E94400601B7A009001D4780175A04F
:1000230000E493F2A308B8000205A0D9F4DAF2754C
:02003300A0FF2C
:1000EC00E508450960467A01E50825E0F508E509CB
:1000FC00334012F509E5829508E58395094003 0A1A
:10010C0080E6C3E50913F509E50813F508C3E58294
:10011C009508F5F0E58395094005F58385F082C3D4
:0D012C00E50913F509E50813F508DAE122ED
:10003B007800E84400600A790175A000E4F309D860
:10004B00FC7800E84400600C7900900001E4F0A318
:04005B00D8FCD9FAFA
:100139007A10E4FBFCE58225E0F582E58333F5835B
:10014900EB33FBEC33FCEB9508F5F0EC9509400635
:09015900FCABF0438201DADD2267
:0D000600758109120 1D0E582600302000033C
:10016200C2D5E58330E70DD2D5E4C39582F582E4AA
:100172009583F583E50930E70BE4C39508F508E4B8
:100182009509F5091200EC30D50BE4C39582F5828E
:10019200E49583F58322C2D5E58330E70DD2D5E419
:1001A200C39582F582E49583F583E50930E70DB2C4
:1001B200D5E4C39508F508E49509F509120139302B
:1001C200D50BE4C39582F582E49583F58322758 28B
:0201D200002209
:00000001FF
```

SDCC REFINED BUILDING EUCLID.C

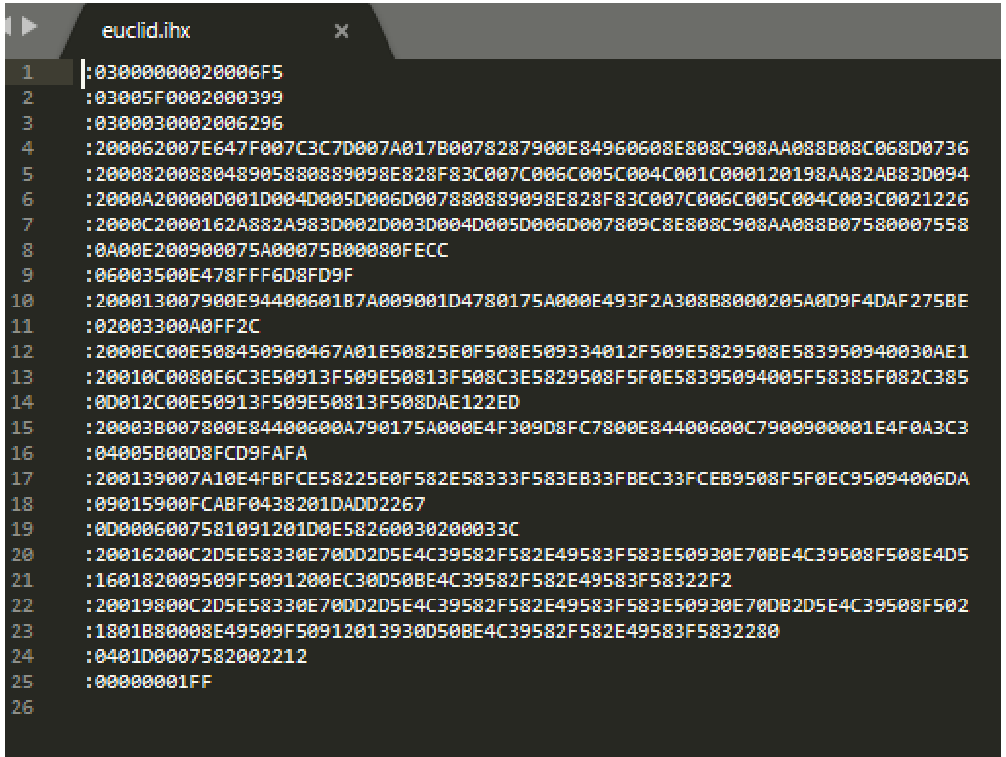
On Command Prompt, from euclid\i8051ISS\euclidSDCCRefined:

sdcc ../../euclid.c –mmcs51

Finally, in euclid\i8051ISS\euclidSDCCRefined folder there should be the following files:



euclid.asm	21/07/2019 19:31	Assembler Source	10 KB
euclid.ihx	21/07/2019 19:31	File IHX	2 KB
euclid.lk	21/07/2019 19:31	File LK	1 KB
euclid.lst	21/07/2019 19:31	MASM Listing	26 KB
euclid.map	21/07/2019 19:31	Documento di testo	19 KB
euclid.mem	21/07/2019 19:31	File MEM	2 KB
euclid.rel	21/07/2019 19:31	File REL	5 KB
euclid.rst	21/07/2019 19:31	File RST	26 KB
euclid.sym	21/07/2019 19:31	File SYM	39 KB



```
:03000000020006F5
:03005F0002000399
:0300030002006296
:200062007E647F007C3C7D007A017B0078287900E84960608E808C908AA088B08C068D0736
:20008200880489058808889098E828F83C007C006C005C004C001C000120198AA82AB83D094
:2000A20000D001D004D005D006D007880889098E828F83C007C006C005C004C003C0021226
:2000C2000162A882A983D002D003D004D005D006D007809C8E808C908AA088B07580007558
:0A00E200900075A00075B00080FECC
:06003500E478FFF6D8FD9F
:200013007900E94400601B7A009001D4780175A000E493F2A308B8000205A0D9F4DAF275BE
:02003300A0FF2C
:2000EC00E508450960467A01E50825E0F508E509334012F509E5829508E58395094003 0AE1
:20010C0080E6C3E50913F509E50813F508C3E5829508F5F0E58395094005F58385F082C385
:0D012C00E50913F509E50813F508DAE122ED
:20003B007800E84400600A790175A000E4F309D8FC7800E84400600C7900900001E4F0A3C3
:04005B00D8FCD9FAFA
:200139007A10E4FBFCE58225E0F582E58333F583EB33FBEC33FCEB9508F5F0EC95094006DA
:09015900FCABF0438201DADD2267
:0D00060075810912 01D0E58260030200033C
:20016200C2D5E58330E70DD2D5E4C39582F582E49583F583E50930E70BE4C39508F508E4D5
:160182009509F5091200EC30D50BE4C39582F582E49583F58322F2
:20019800C2D5E58330E70DD2D5E4C39582F582E49583F583E50930E70DB2D5E4C39508F502
:1801B80008E49509F50912013930D50BE4C39582F582E49583F5832280
:0401D00075820022 12
:00000001FF

```

SDCC REFINED BUILDING EUCLID.C

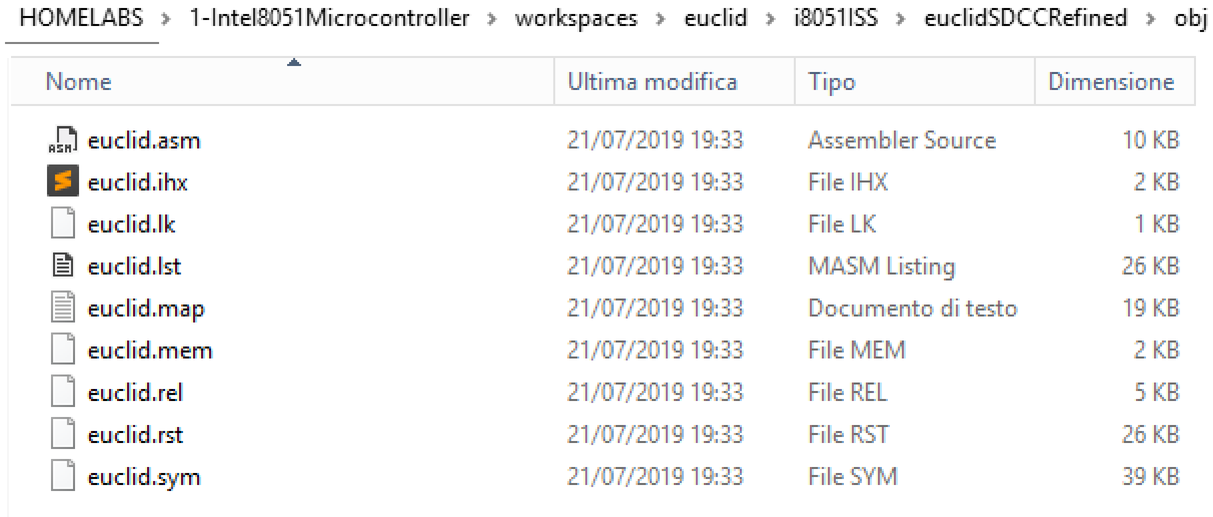
I created obj folder in euclid\i8051ISS\euclidSDCCRefined.

On Command Prompt, from euclid\i8051ISS\euclidSDCCRefined:

- To have a better refinement (considering ram size):

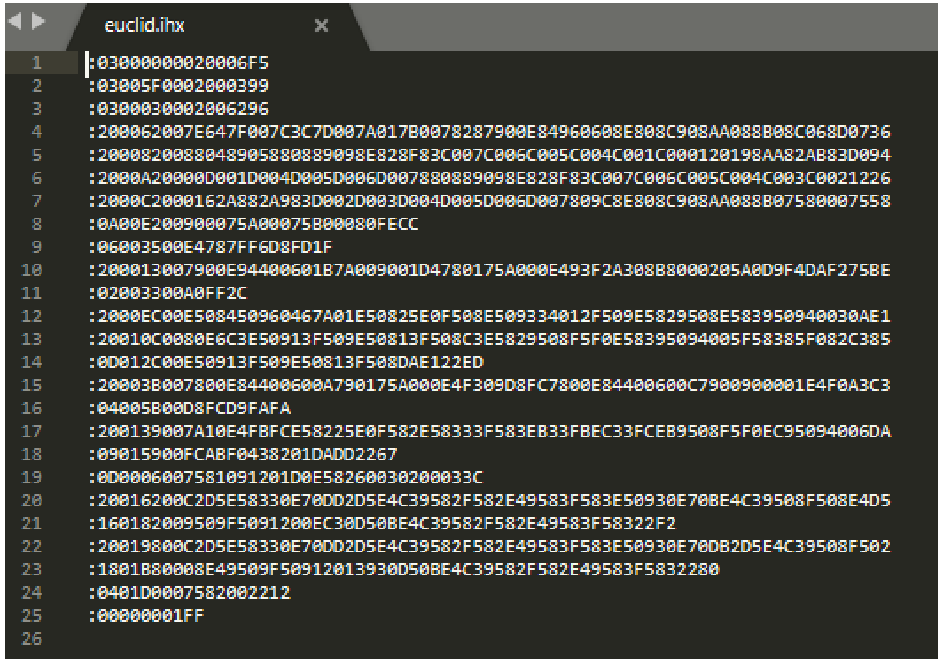
sdcc ../../euclid.c –mmcs51 -–iram-size 128 –o ./obj/

Finally, in euclid\i8051ISS\euclidSDCCRefined\obj folder there should be the following files:



HOMELABS › 1-Intel8051Microcontroller › workspaces › euclid › i8051ISS › euclidSDCCRefined › obj

Nome	Ultima modifica	Tipo	Dimensione
euclid.asm	21/07/2019 19:33	Assembler Source	10 KB
euclid.ihx	21/07/2019 19:33	File IHX	2 KB
euclid.lk	21/07/2019 19:33	File LK	1 KB
euclid.lst	21/07/2019 19:33	MASM Listing	26 KB
euclid.map	21/07/2019 19:33	Documento di testo	19 KB
euclid.mem	21/07/2019 19:33	File MEM	2 KB
euclid.rel	21/07/2019 19:33	File REL	5 KB
euclid.rst	21/07/2019 19:33	File RST	26 KB
euclid.sym	21/07/2019 19:33	File SYM	39 KB



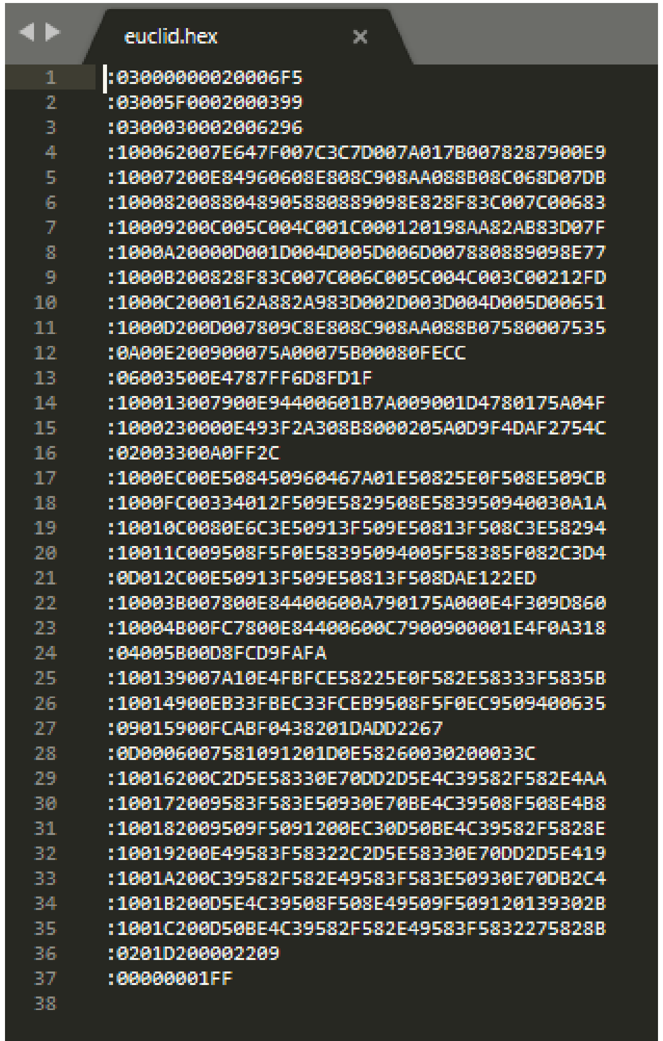
```
:03000000020006F5
:03005F0002000399
:0300030002006296
:200062007E647F007C3C7D007A017B0078287900E84960608E808C908AA088B08C068D0736
:20008200880489058808889098E828F83C007C006C005C004C001C000120198AA82AB83D094
:2000A20000D001D004D005D006D007880889098E828F83C007C006C005C004C003C0021226
:2000C2000162A882A983D002D003D004D005D006D007809C8E808C908AA088B07580007558
:0A00E200900075A00075B00080FECC
:06003500E4787FF6D8FD1F
:200013007900E94400601B7A009001D4780175A000E493F2A308B8000205A0D9F4DAF275BE
:02003300A0FF2C
:2000EC00E508450960467A01E50825E0F508E509334012F509E5829508E58395094003AE1
:20010C0080E6C3E50913F509E50813F508C3E5829508F5F0E58395094005F58385F082C385
:0D012C00E50913F509E50813F508DAE122ED
:20003B007800E84400600A790175A000E4F309D8FC7800E84400600C7900900001E4F0A3C3
:04005B00D8FCD9FAFA
:200139007A10E4FBFCE58225E0F582E58333F583EB33FBEC33FCEB9508F5F0EC95094006DA
:09015900FCABF0438201DADD2267
:0D000600758109120100E582600302000033C
:20016200C2D5E58330E70DD2D5E4C39582F582E49583F583E50930E70BE4C39508F508E4D5
:160182009509F5091200EC30D50BE4C39582F582E49583F58322F2
:20019800C2D5E58330E70DD2D5E4C39582F582E49583F583E50930E70DB2D5E4C39508F502
:1801B80008E49509F50912013930D50BE4C39582F582E49583F5832280
:0401D00007582002212
:00000001FF
```

SDCC REFINED BUILDING EUCLID.C

On Command Prompt, from euclid\i8051ISS\euclidSDCCRefined:

packihx ./obj/euclid.ihx **>** **.**/obj/euclid.hex

Finally, euclid.hex file should have been created from euclid.ihx.

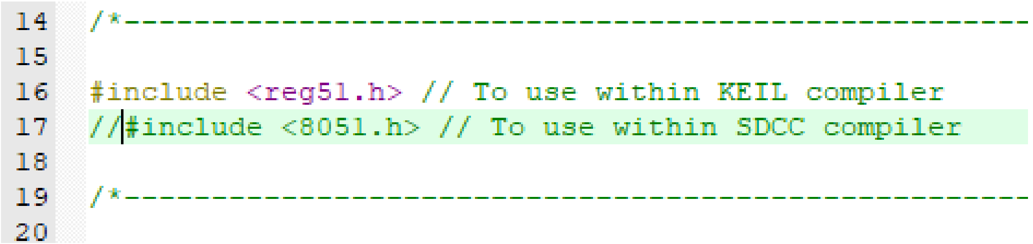


```
:03000000020006F5
:03005F0002000399
:0300030002006296
:100062007E647F007C3C7D007A017B0078287900E9
:10007200E84960608E808C908AA088B08C068D07DB
:1000820088804890588088909 8E828F83C007C00683
:10009200C005C004C001C000120198AA82AB83D07F
:1000A20000D001D004D005D006D00788088909 8E77
:1000B200828F83C007C006C005C004C003C00212FD
:1000C2000162A882A983D002D003D004D005D00651
:1000D200D007809C8E808C908AA088B07580007535
:0A00E200900075A00075B00080FECC
:06003500E4787FF6D8FD1F
:100013007900E94400601B7A009001D4780175A04F
:1000230000E493F2A308B8000205A0D9F4DAF2754C
:02003300A0FF2C
:1000EC00E508450960467A01E50825E0F508E509CB
:1000FC00334012F509E5829508E583950940030A1A
:10010C0080E6C3E50913F509E50813F508C3E58294
:10011C009508F5F0E583950940 05F58385F082C3D4
:0D012C00E50913F509E50813F508DAE122ED
:10003B007800E84400600A790175A000E4F309D860
:10004B00FC7800E84400600C7900900001E4F0A318
:04005B00D8FCD9FAFA
:100139007A10E4FBFCE58225E0F582E58333F5835B
:10014900EB33FBEC33FCEB9508F5F0EC950940 0635
:09015900FCABF04382 01DADD2267
:0D00060075810912 01D0E582600302 00033C
:10016200C2D5E58330E70DD2D5E4C39582F582E4AA
:100172009583F583E50930E70BE4C39508F508E4B8
:100182009509F5091200EC30D50BE4C39582F5828E
:10019200E49583F58322C2D5E58330E70DD2D5E419
:1001A200C39582F582E49583F583E50930E70DB2C4
:1001B200D5E4C39508F508E49509F50912 0139302B
:1001C200D50BE4C39582F582E49583F58322 75828B
:0201D200002209
:00000001FF
```

KEIL BUILDING EUCLID.C

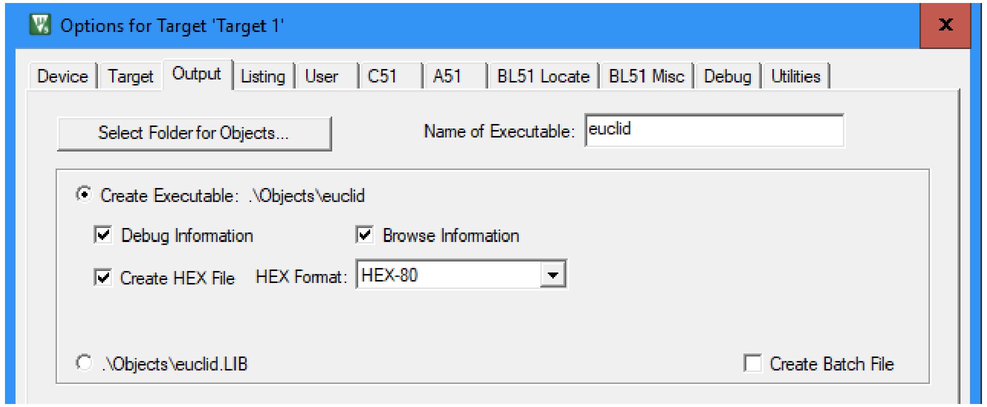
I created a i8051 Keil progect in euclid\i8051ISS\euclidKeil, adding euclid.c file to it.

I changed dependencies for Keil compiler:



```
/*--------------------------------------------------

#include <reg51.h> // To use within KEIL compiler
//#include <8051.h> // To use within SDCC compiler

/*--------------------------------------------------
```

I configured Keil output:



Name of Executable: euclid

● Create Executable: .\Objects\euclid

☑ Debug Information ☑ Browse Information

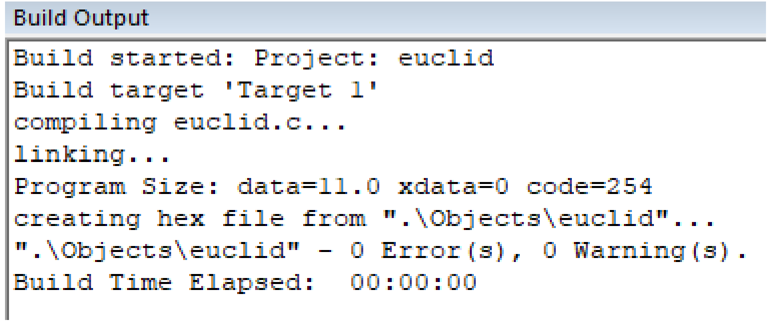
☑ Create HEX File HEX Format: HEX-80

○ .\Objects\euclid.LIB

□ Create Batch File

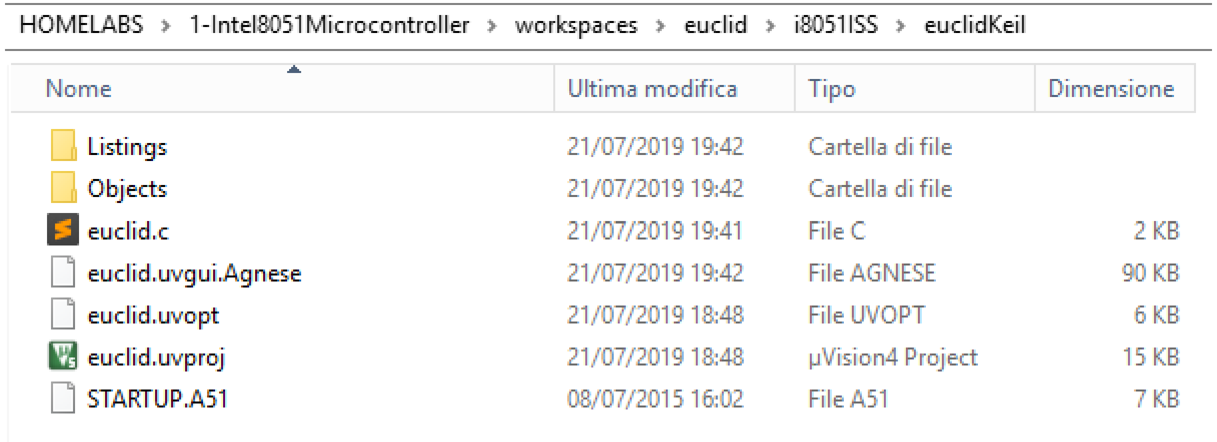
KEIL BUILDING EUCLID.C

Then, I built to target:



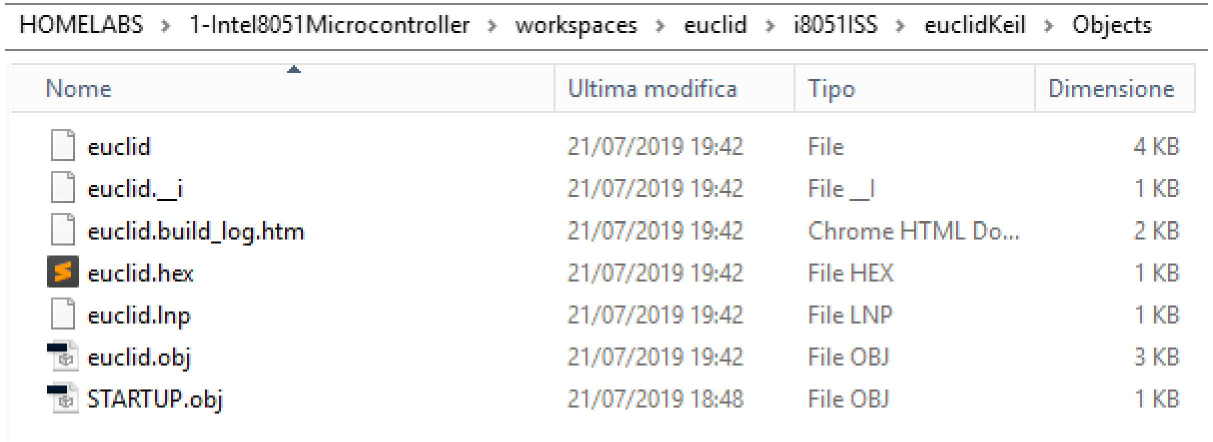
Build Output

```
Build started: Project: euclid
Build target 'Target 1'
compiling euclid.c...
linking...
Program Size: data=11.0 xdata=0 code=254
creating hex file from ".\Objects\euclid"...
".\Objects\euclid" - 0 Error(s), 0 Warning(s).
Build Time Elapsed:  00:00:00
```



HOMELABS › 1-Intel8051Microcontroller › workspaces › euclid › i8051ISS › euclidKeil

Nome	Ultima modifica	Tipo	Dimensione
Listings	21/07/2019 19:42	Cartella di file	
Objects	21/07/2019 19:42	Cartella di file	
euclid.c	21/07/2019 19:41	File C	2 KB
euclid.uvgui.Agnese	21/07/2019 19:42	File AGNESE	90 KB
euclid.uvopt	21/07/2019 18:48	File UVOPT	6 KB
euclid.uvproj	21/07/2019 18:48	µVision4 Project	15 KB
STARTUP.A51	08/07/2015 16:02	File A51	7 KB



HOMELABS › 1-Intel8051Microcontroller › workspaces › euclid › i8051ISS › euclidKeil › Objects

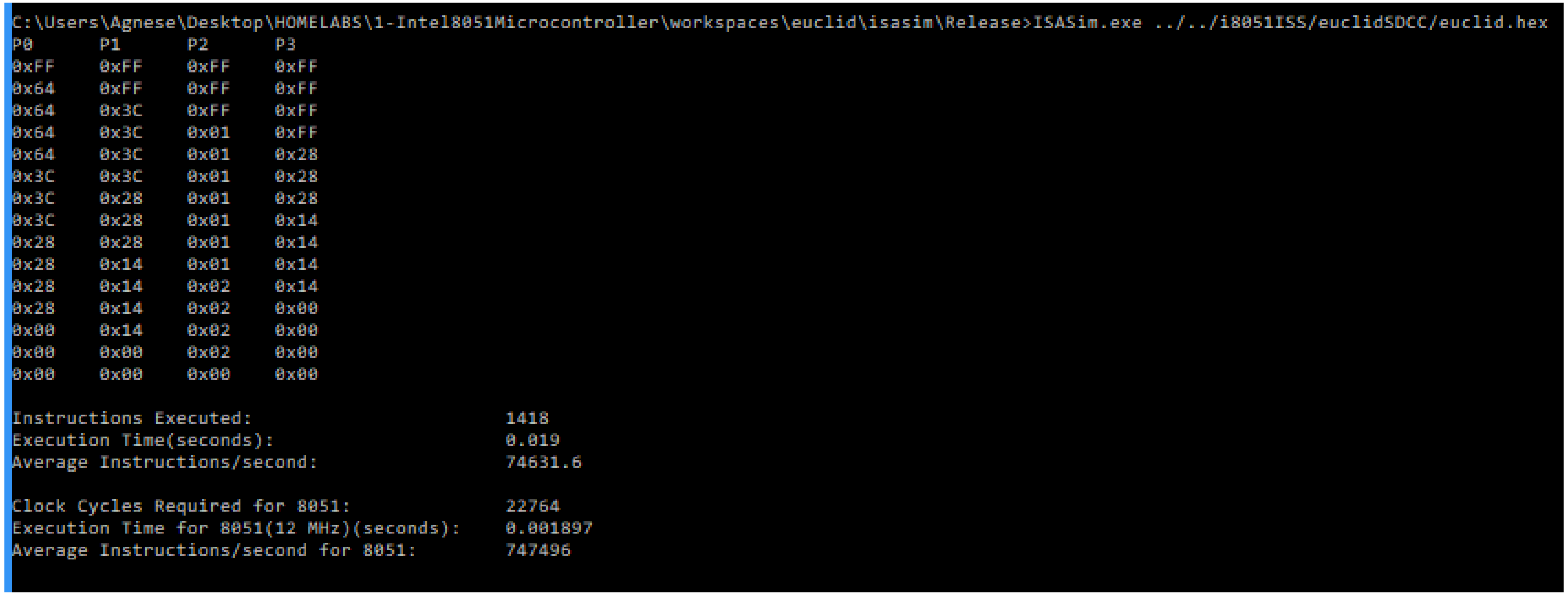
Nome	Ultima modifica	Tipo	Dimensione
euclid	21/07/2019 19:42	File	4 KB
euclid.__i	21/07/2019 19:42	File __I	1 KB
euclid.build_log.htm	21/07/2019 19:42	Chrome HTML Do...	2 KB
euclid.hex	21/07/2019 19:42	File HEX	1 KB
euclid.lnp	21/07/2019 19:42	File LNP	1 KB
euclid.obj	21/07/2019 19:42	File OBJ	3 KB
STARTUP.obj	21/07/2019 18:48	File OBJ	1 KB

ISS SIMULATION - SDCC

I simulated via ISASim.exe the .hex file, previously obtained by SDCC compiler.

On Command Prompt, from euclid\isasim\Release:

ISASim.exe ../../i8051ISS/euclidSDCC/euclid.hex



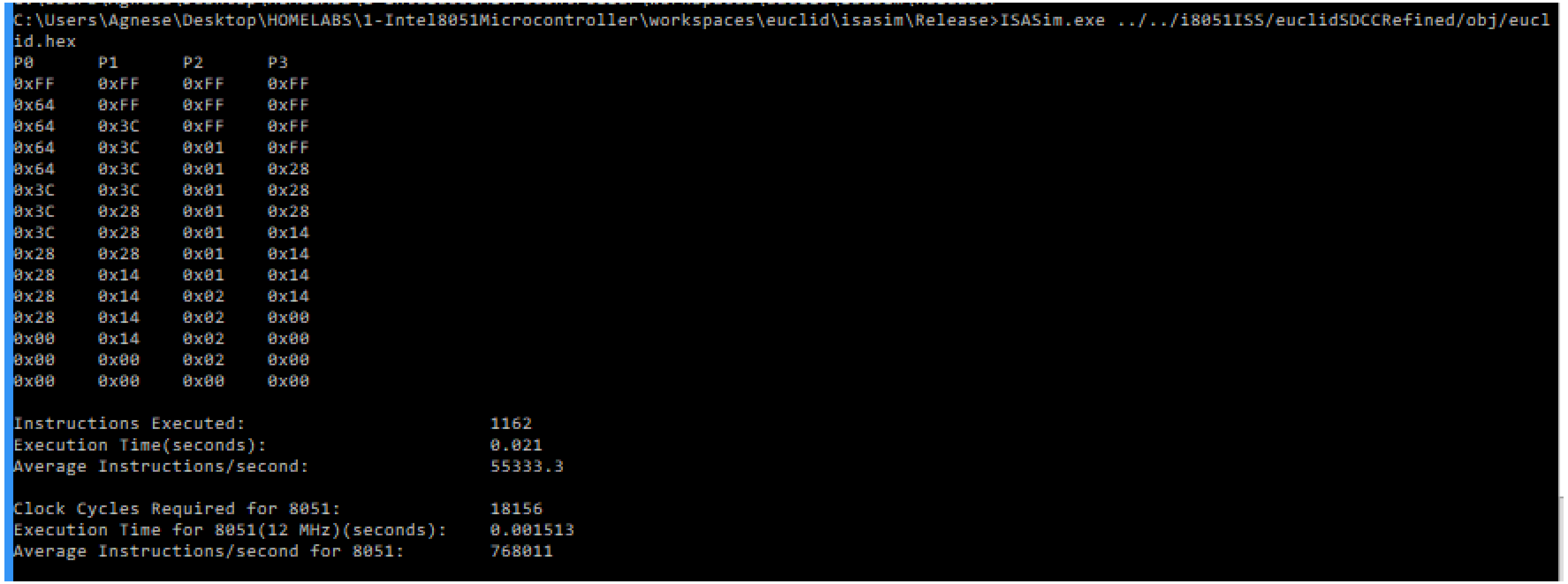
```
C:\Users\Agnese\Desktop\HOMELABS\1-Intel8051Microcontroller\workspaces\euclid\isasim\Release>ISASim.exe ../../i8051ISS/euclidSDCC/euclid.hex
P0      P1      P2      P3
0xFF    0xFF    0xFF    0xFF
0x64    0xFF    0xFF    0xFF
0x64    0x3C    0xFF    0xFF
0x64    0x3C    0x01    0xFF
0x64    0x3C    0x01    0x28
0x3C    0x3C    0x01    0x28
0x3C    0x28    0x01    0x28
0x3C    0x28    0x01    0x14
0x28    0x28    0x01    0x14
0x28    0x14    0x01    0x14
0x28    0x14    0x02    0x14
0x28    0x14    0x02    0x00
0x00    0x14    0x02    0x00
0x00    0x00    0x02    0x00
0x00    0x00    0x00    0x00

Instructions Executed:                      1418
Execution Time(seconds):                    0.019
Average Instructions/second:                74631.6

Clock Cycles Required for 8051:             22764
Execution Time for 8051(12 MHz)(seconds):   0.001897
Average Instructions/second for 8051:       747496
```

ISS SIMULATION – SDCC REFINED

I simulated via ISASim.exe the .hex file, previously obtained by SDCC Refined compiler.

On Command Prompt, from euclid\isasim\Release:

ISASim.exe ../../i8051ISS/euclidSDCCRefined/obj/euclid.hex



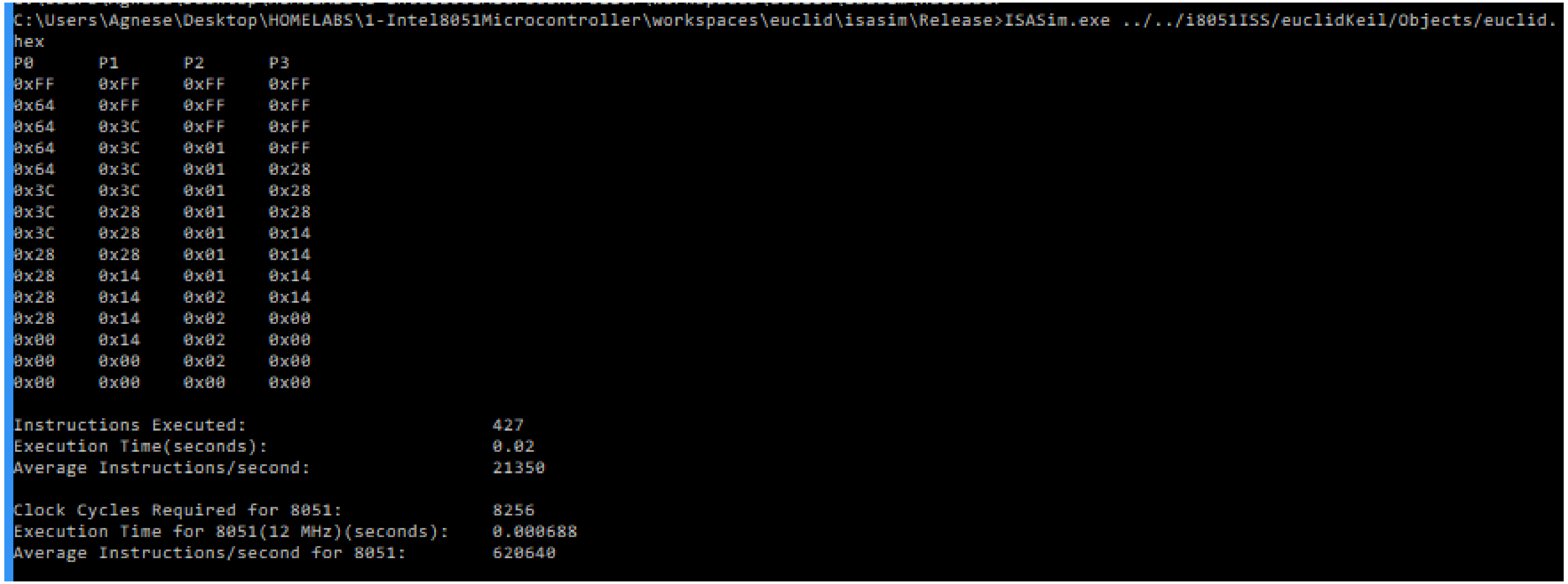
```
C:\Users\Agnese\Desktop\HOMELABS\1-Intel8051Microcontroller\workspaces\euclid\isasim\Release>ISASim.exe ../../i8051ISS/euclidSDCCRefined/obj/eucl
id.hex
P0      P1      P2      P3
0xFF    0xFF    0xFF    0xFF
0x64    0xFF    0xFF    0xFF
0x64    0x3C    0xFF    0xFF
0x64    0x3C    0x01    0xFF
0x64    0x3C    0x01    0x28
0x3C    0x3C    0x01    0x28
0x3C    0x28    0x01    0x28
0x3C    0x28    0x01    0x14
0x28    0x28    0x01    0x14
0x28    0x14    0x01    0x14
0x28    0x14    0x02    0x14
0x28    0x14    0x02    0x00
0x00    0x14    0x02    0x00
0x00    0x00    0x02    0x00
0x00    0x00    0x00    0x00

Instructions Executed:                      1162
Execution Time(seconds):                    0.021
Average Instructions/second:                55333.3

Clock Cycles Required for 8051:             18156
Execution Time for 8051(12 MHz)(seconds):   0.001513
Average Instructions/second for 8051:       768011
```

ISS SIMULATION – KEIL

I simulated via ISASim.exe the .hex file, previously obtained by Keil compiler.

On Command Prompt, from euclid\isasim\Release:

ISASim.exe ../../i8051ISS/euclidKeil/Objects/euclid.hex

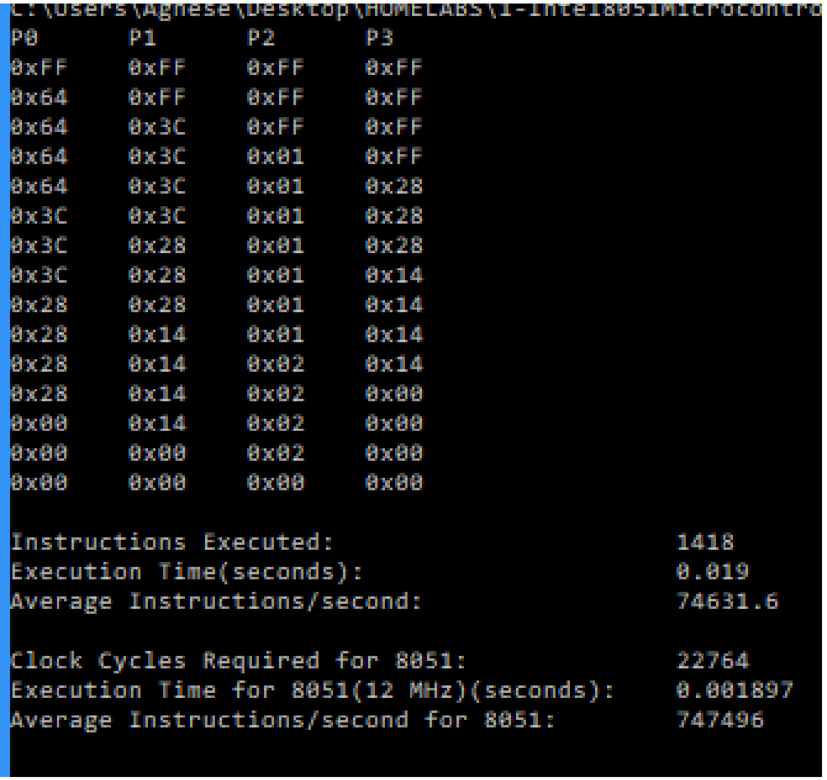


```
C:\Users\Agnese\Desktop\HOMELABS\1-Intel8051Microcontroller\workspaces\euclid\isasim\Release>ISASim.exe ../../i8051ISS/euclidKeil/Objects/euclid.
hex
P0      P1      P2      P3
0xFF    0xFF    0xFF    0xFF
0x64    0xFF    0xFF    0xFF
0x64    0x3C    0xFF    0xFF
0x64    0x3C    0x01    0xFF
0x64    0x3C    0x01    0x28
0x3C    0x3C    0x01    0x28
0x3C    0x28    0x01    0x28
0x3C    0x28    0x01    0x14
0x28    0x28    0x01    0x14
0x28    0x14    0x01    0x14
0x28    0x14    0x02    0x14
0x28    0x14    0x02    0x00
0x00    0x14    0x02    0x00
0x00    0x00    0x02    0x00
0x00    0x00    0x00    0x00

Instructions Executed:                      427
Execution Time(seconds):                    0.02
Average Instructions/second:                21350

Clock Cycles Required for 8051:             8256
Execution Time for 8051(12 MHz)(seconds):   0.000688
Average Instructions/second for 8051:       620640
```

ISS SIMULATION – COMPILERS COMPARISON

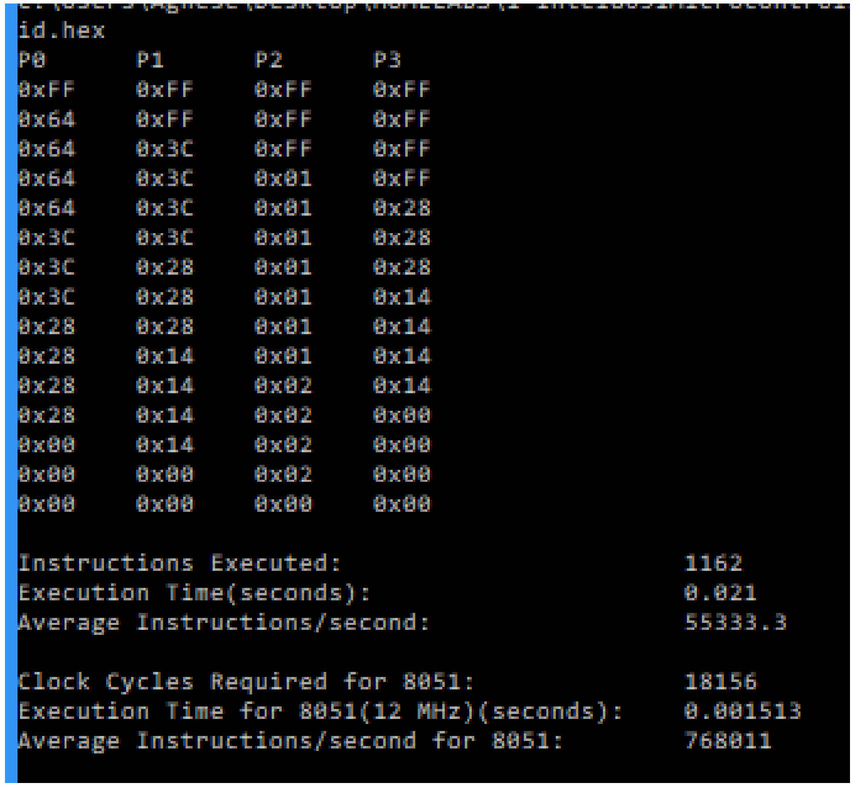
SDCC Refined is better than SDCC, but Keil is the best.

SDCC



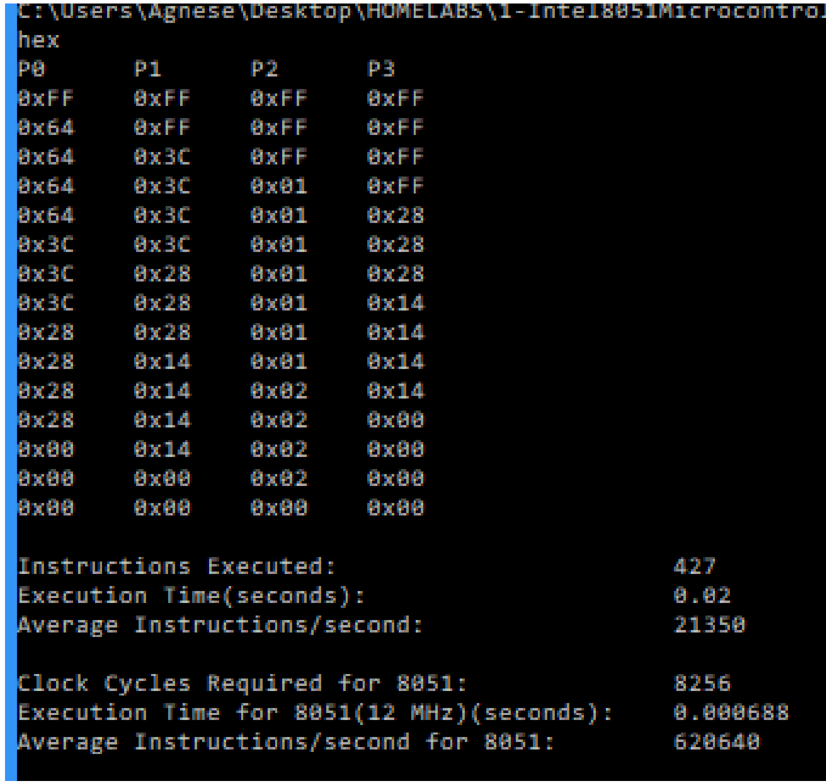
```
C:\Users\Agnese\Desktop\HOMELABS\1-Intel8051Microcontro
P0      P1      P2      P3
0xFF    0xFF    0xFF    0xFF
0x64    0xFF    0xFF    0xFF
0x64    0x3C    0xFF    0xFF
0x64    0x3C    0x01    0xFF
0x64    0x3C    0x01    0x28
0x3C    0x3C    0x01    0x28
0x3C    0x28    0x01    0x28
0x3C    0x28    0x01    0x14
0x28    0x28    0x01    0x14
0x28    0x14    0x01    0x14
0x28    0x14    0x02    0x14
0x28    0x14    0x02    0x00
0x00    0x14    0x02    0x00
0x00    0x00    0x02    0x00
0x00    0x00    0x00    0x00

Instructions Executed:                          1418
Execution Time(seconds):                        0.019
Average Instructions/second:                    74631.6

Clock Cycles Required for 8051:                 22764
Execution Time for 8051(12 MHz)(seconds):       0.001897
Average Instructions/second for 8051:           747496
```

SDCC Refined



```
C:\Users\Agnese\Desktop\HOMELABS\1-Intel8051Microcontrol
id.hex
P0      P1      P2      P3
0xFF    0xFF    0xFF    0xFF
0x64    0xFF    0xFF    0xFF
0x64    0x3C    0xFF    0xFF
0x64    0x3C    0x01    0xFF
0x64    0x3C    0x01    0x28
0x3C    0x3C    0x01    0x28
0x3C    0x28    0x01    0x28
0x3C    0x28    0x01    0x14
0x28    0x28    0x01    0x14
0x28    0x14    0x01    0x14
0x28    0x14    0x02    0x14
0x28    0x14    0x02    0x00
0x00    0x14    0x02    0x00
0x00    0x00    0x02    0x00
0x00    0x00    0x00    0x00

Instructions Executed:                          1162
Execution Time(seconds):                        0.021
Average Instructions/second:                    55333.3

Clock Cycles Required for 8051:                 18156
Execution Time for 8051(12 MHz)(seconds):       0.001513
Average Instructions/second for 8051:           768011
```

Keil



```
C:\Users\Agnese\Desktop\HOMELABS\1-Intel8051Microcontrol
hex
P0      P1      P2      P3
0xFF    0xFF    0xFF    0xFF
0x64    0xFF    0xFF    0xFF
0x64    0x3C    0xFF    0xFF
0x64    0x3C    0x01    0xFF
0x64    0x3C    0x01    0x28
0x3C    0x3C    0x01    0x28
0x3C    0x28    0x01    0x28
0x3C    0x28    0x01    0x14
0x28    0x28    0x01    0x14
0x28    0x14    0x01    0x14
0x28    0x14    0x02    0x14
0x28    0x14    0x02    0x00
0x00    0x14    0x02    0x00
0x00    0x00    0x02    0x00
0x00    0x00    0x00    0x00

Instructions Executed:                          427
Execution Time(seconds):                        0.02
Average Instructions/second:                    21350

Clock Cycles Required for 8051:                 8256
Execution Time for 8051(12 MHz)(seconds):       0.000688
Average Instructions/second for 8051:           620640
```

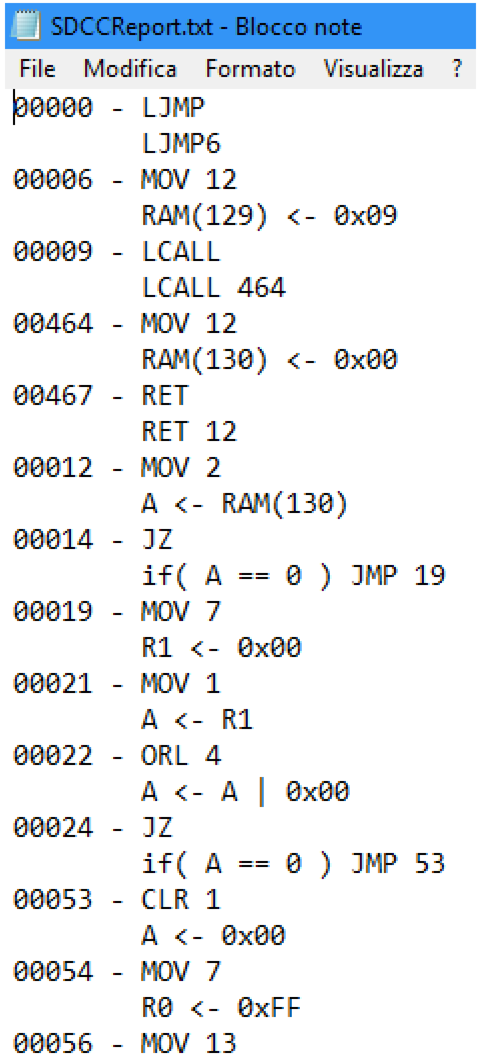
ISS RELEASE FOR DEBUGGING SIMULATION

I simulated via Release For Debugging ISASim.exe the .hex files, previously obtained by SDCC, SDCC Refined and Keil compilers.

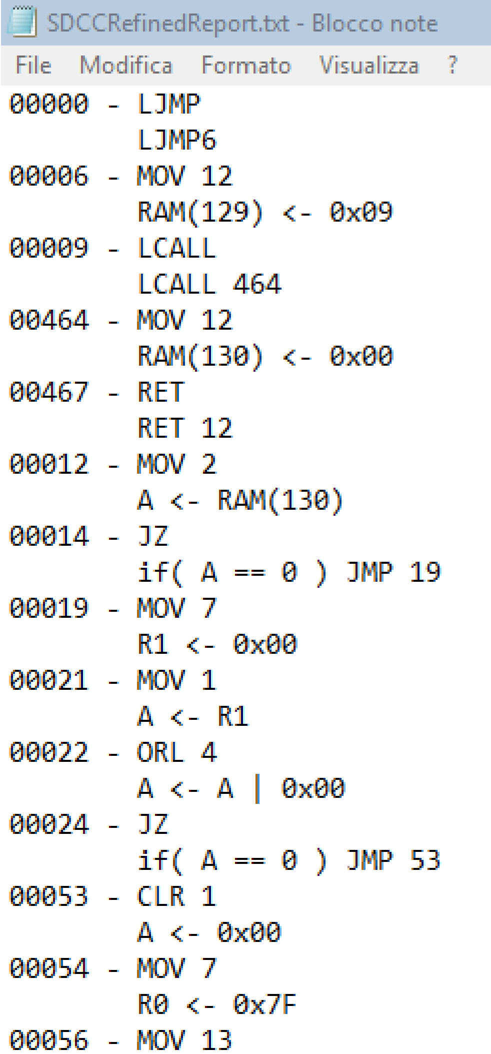
On Command Prompt, from euclid\isasim\ReleaseForDebugging:

- ISASim.exe ../../i8051ISS/euclidSDCC/euclid.hex ../../i8051ISS/euclidSDCC/*SDCCReport*.txt
- ISASim.exe ../../i8051ISS/euclidSDCCRefined/obj/euclid.hex ../../i8051ISS/euclidSDCCRefined/obj/SDCCRefinedReport.txt
- ISASim.exe ../../i8051ISS/euclidKeil/Objects/euclid.hex ../../i8051ISS/euclidKeil/Objects/Keil*Report*.txt

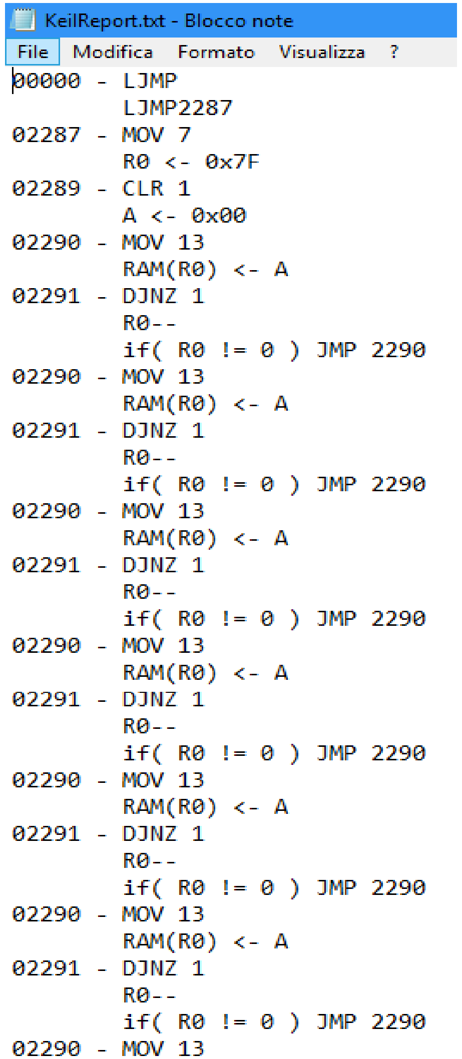
ISS RELEASE FOR DEBUGGING SIMULATION – OUTPUT FILES



```
00000 - LJMP
        LJMP6
00006 - MOV 12
        RAM(129) <- 0x09
00009 - LCALL
        LCALL 464
00464 - MOV 12
        RAM(130) <- 0x00
00467 - RET
        RET 12
00012 - MOV 2
        A <- RAM(130)
00014 - JZ
        if( A == 0 ) JMP 19
00019 - MOV 7
        R1 <- 0x00
00021 - MOV 1
        A <- R1
00022 - ORL 4
        A <- A | 0x00
00024 - JZ
        if( A == 0 ) JMP 53
00053 - CLR 1
        A <- 0x00
00054 - MOV 7
        R0 <- 0xFF
00056 - MOV 13
```



```
00000 - LJMP
        LJMP6
00006 - MOV 12
        RAM(129) <- 0x09
00009 - LCALL
        LCALL 464
00464 - MOV 12
        RAM(130) <- 0x00
00467 - RET
        RET 12
00012 - MOV 2
        A <- RAM(130)
00014 - JZ
        if( A == 0 ) JMP 19
00019 - MOV 7
        R1 <- 0x00
00021 - MOV 1
        A <- R1
00022 - ORL 4
        A <- A | 0x00
00024 - JZ
        if( A == 0 ) JMP 53
00053 - CLR 1
        A <- 0x00
00054 - MOV 7
        R0 <- 0x7F
00056 - MOV 13
```



```
00000 - LJMP
        LJMP2287
02287 - MOV 7
        R0 <- 0x7F
02289 - CLR 1
        A <- 0x00
02290 - MOV 13
        RAM(R0) <- A
02291 - DJNZ 1
        R0--
        if( R0 != 0 ) JMP 2290
02290 - MOV 13
        RAM(R0) <- A
02291 - DJNZ 1
        R0--
        if( R0 != 0 ) JMP 2290
02290 - MOV 13
        RAM(R0) <- A
02291 - DJNZ 1
        R0--
        if( R0 != 0 ) JMP 2290
02290 - MOV 13
        RAM(R0) <- A
02291 - DJNZ 1
        R0--
        if( R0 != 0 ) JMP 2290
02290 - MOV 13
        RAM(R0) <- A
02291 - DJNZ 1
        R0--
        if( R0 != 0 ) JMP 2290
02290 - MOV 13
        RAM(R0) <- A
02291 - DJNZ 1
        R0--
        if( R0 != 0 ) JMP 2290
02290 - MOV 13
```

HEX2ROM BUILDING SDCC HEX FILES

I generated the .vhd rom file from every compiled .hex file.

On Command Prompt, from euclid\hex2rom\Release:

- To change EOL (End Of File) as needed by Hex2Rom.exe:

```
tr -d '\015' < ../../i8051ISS/euclidSDCC/euclid.hex > ../../i8051ISS/euclidSDCC/euclid.hex_EOL.hex
```

- To generate rom file:

```
Hex2Rom.exe ../../i8051ISS/euclidSDCC/euclid.hex_EOL.hex
```

The resulting file is i8051_rom.vhd. I moved it into euclid\i8051VHDL\i8051SDCC\model (overvriting the existing file).

HEX2ROM BUILDING SDCC REFINED HEX FILES

With Command Prompt, from euclid\hex2rom\Release:

- To change EOL (End Of File) as needed by Hex2Rom.exe:

```
tr -d '\015' < ../../i8051ISS/euclidSDCCRefined/obj/euclid.hex > ../../i8051ISS/euclidSDCCRefined/obj/euclid.hex_EOL.hex
```

- To generate rom file:

```
Hex2Rom.exe ../../i8051ISS/euclidSDCCRefined/obj/euclid.hex_EOL.hex
```

The resulting file is i8051_rom.vhd. I moved it into euclid\i8051VHDL\i8051SDCCRefined\model (overvriting the existing file).

HEX2ROM BUILDING KEIL HEX FILES

First of all, I generated the .vhd rom file from every compiled .hex file.

With Command Prompt, from euclid\hex2rom\Release:

- To change EOL (End Of File) as needed by Hex2Rom.exe:

```
tr -d '\015' < ../../i8051ISS/euclidKeil/Objects/euclid.hex > ../../i8051ISS/euclidKeil/Objects/euclid.hex_EOL.hex
```

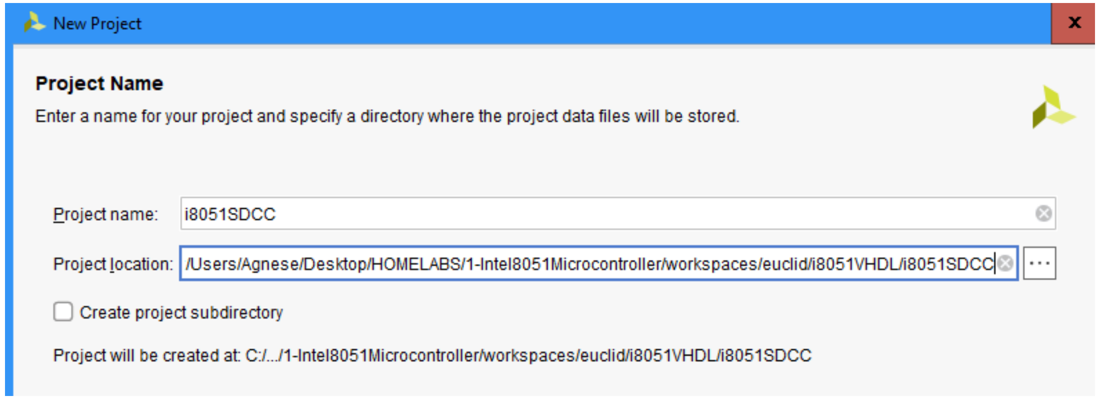
- To generate rom file:

```
Hex2Rom.exe ../../i8051ISS/euclidKeil/Objects/euclid.hex_EOL.hex
```

The resulting file is i8051_rom.vhd. I moved it into euclid\i8051VHDL\i8051SDCCRefined\model (overvriting the existing file).

VIVADO PROJECTS CREATION

I created an Artix7 Vivado Project for every compiled .hex file in euclid\i8051VHDL, by making a new project in correspondence with each previously created euclid\i8051VHDL subfolder and adding the files contained in model subfolder to it.



Project Name

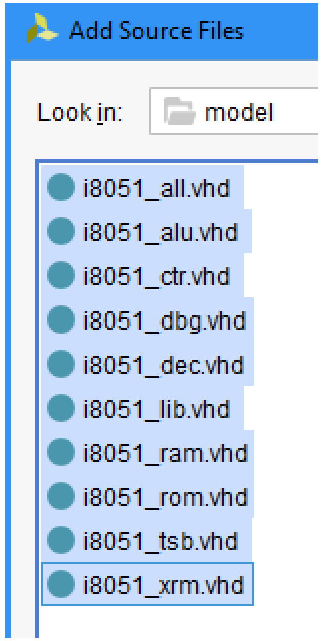
Enter a name for your project and specify a directory where the project data files will be stored.

Project name: i8051SDCC

Project location: /Users/Agnese/Desktop/HOMELABS/1-Intel8051Microcontroller/workspaces/euclid/i8051VHDL/i8051SDCC

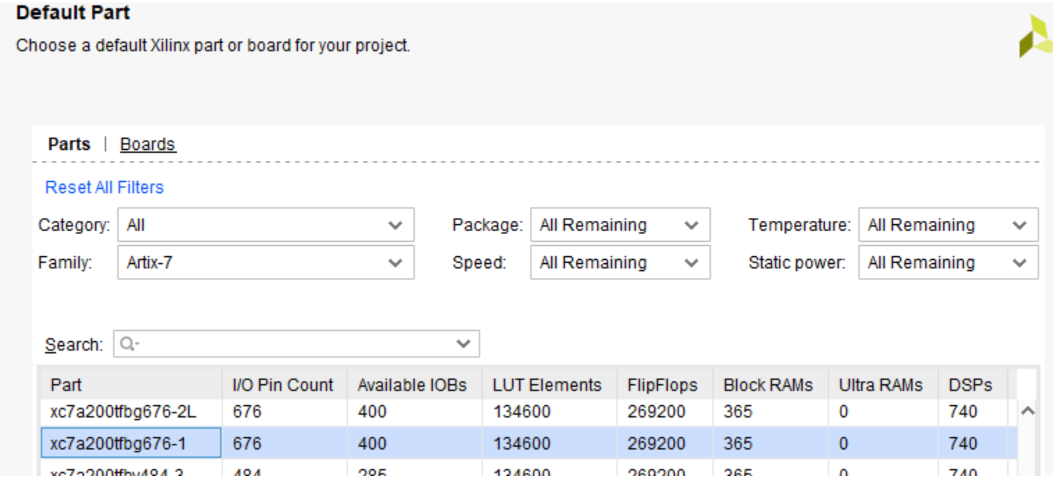
□ Create project subdirectory

Project will be created at: C:/.../1-Intel8051Microcontroller/workspaces/euclid/i8051VHDL/i8051SDCC



Look in: model

- i8051_all.vhd
- i8051_alu.vhd
- i8051_ctr.vhd
- i8051_dbg.vhd
- i8051_dec.vhd
- i8051_lib.vhd
- i8051_ram.vhd
- i8051_rom.vhd
- i8051_tsb.vhd
- i8051_xrm.vhd


Default Part

Choose a default Xilinx part or board for your project.

Category: All

Family: Artix-7

Package: All Remaining

Speed: All Remaining

Temperature: All Remaining

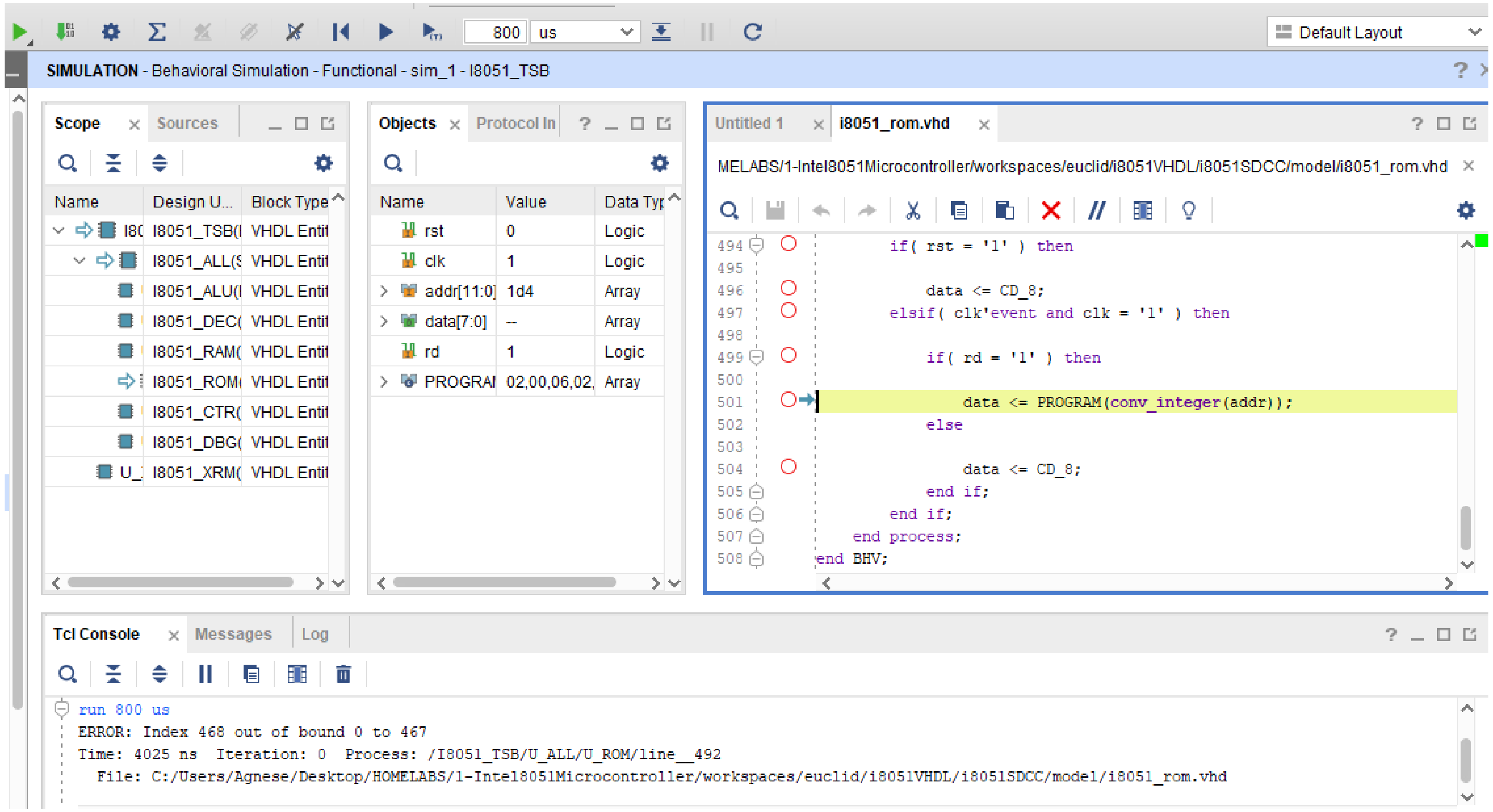
Static power: All Remaining

Search:

Part	I/O Pin Count	Available IOBs	LUT Elements	FlipFlops	Block RAMs	Ultra RAMs	DSPs
xc7a200tfbg676-2L	676	400	134600	269200	365	0	740
xc7a200tfbg676-1	676	400	134600	269200	365	0	740
xc7a200tfbv484-3	484	285	134600	269200	365	0	740

VHDL SDCC SIMULATION

The file compiled by SDCC generates an out of bound error, because it ignores memory size contraints.



SIMULATION - Behavioral Simulation - Functional - sim_1 - I8051_TSB

Name	Design U...	Block Type
I80	I8051_TSB(	VHDL Entit
	I8051_ALL(	VHDL Entit
	I8051_ALU(	VHDL Entit
	I8051_DEC(	VHDL Entit
	I8051_RAM(	VHDL Entit
	I8051_ROM(	VHDL Entit
	I8051_CTR(	VHDL Entit
	I8051_DBG(	VHDL Entit
U_	I8051_XRM(	VHDL Entit

Name	Value	Data Typ
rst	0	Logic
clk	1	Logic
addr[11:0]	1d4	Array
data[7:0]	--	Array
rd	1	Logic
PROGRAI	02,00,06,02,	Array

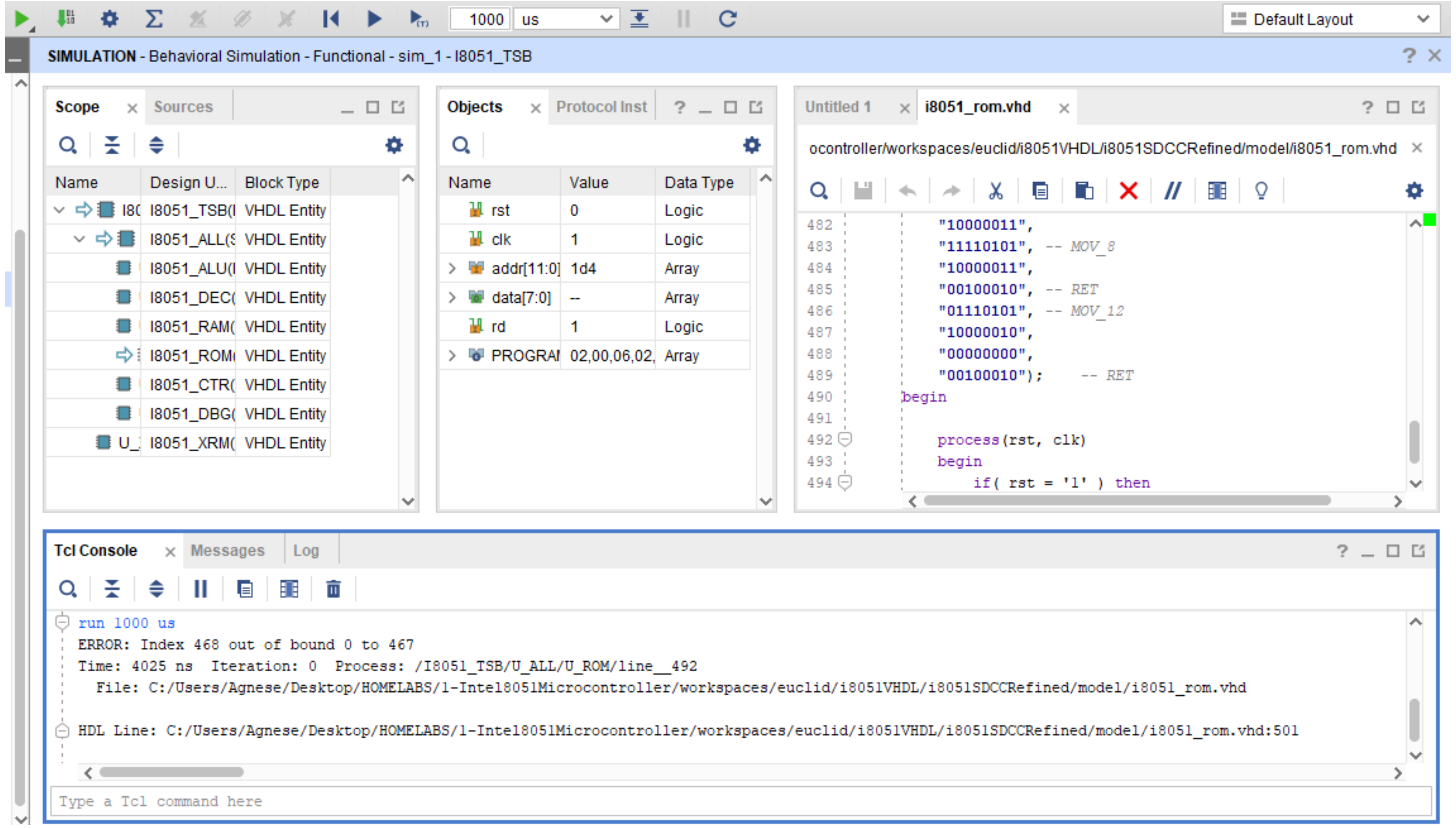
MELABS/1-Intel8051Microcontroller/workspaces/euclid/i8051VHDL/i8051SDCC/model/i8051_rom.vhd

```
        if( rst = '1' ) then

            data <= CD_8;
        elsif( clk'event and clk = '1' ) then

            if( rd = '1' ) then

                data <= PROGRAM(conv_integer(addr));
            else

                data <= CD_8;
            end if;
        end if;
    end process;
end BHV;
```

```
run 800 us
ERROR: Index 468 out of bound 0 to 467
Time: 4025 ns  Iteration: 0  Process: /I8051_TSB/U_ALL/U_ROM/line__492
  File: C:/Users/Agnese/Desktop/HOMELABS/1-Intel8051Microcontroller/workspaces/euclid/i8051VHDL/i8051SDCC/model/i8051_rom.vhd
```

VHDL SDCC REFINED SIMULATION

The file compiled by SDCC Refined generates an out of bound error too, even if it should't because it takes into account memory size contraints (I tried again and again in search of errors).



SIMULATION - Behavioral Simulation - Functional - sim_1 - I8051_TSB

Name	Design U...	Block Type
I8(	I8051_TSB(I	VHDL Entity
	I8051_ALL(S	VHDL Entity
	I8051_ALU(I	VHDL Entity
	I8051_DEC(	VHDL Entity
	I8051_RAM(	VHDL Entity
	I8051_ROM(	VHDL Entity
	I8051_CTR(	VHDL Entity
	I8051_DBG(	VHDL Entity
U_	I8051_XRM(	VHDL Entity

Name	Value	Data Type
rst	0	Logic
clk	1	Logic
addr[11:0]	1d4	Array
data[7:0]	--	Array
rd	1	Logic
PROGRAI	02,00,06,02,	Array

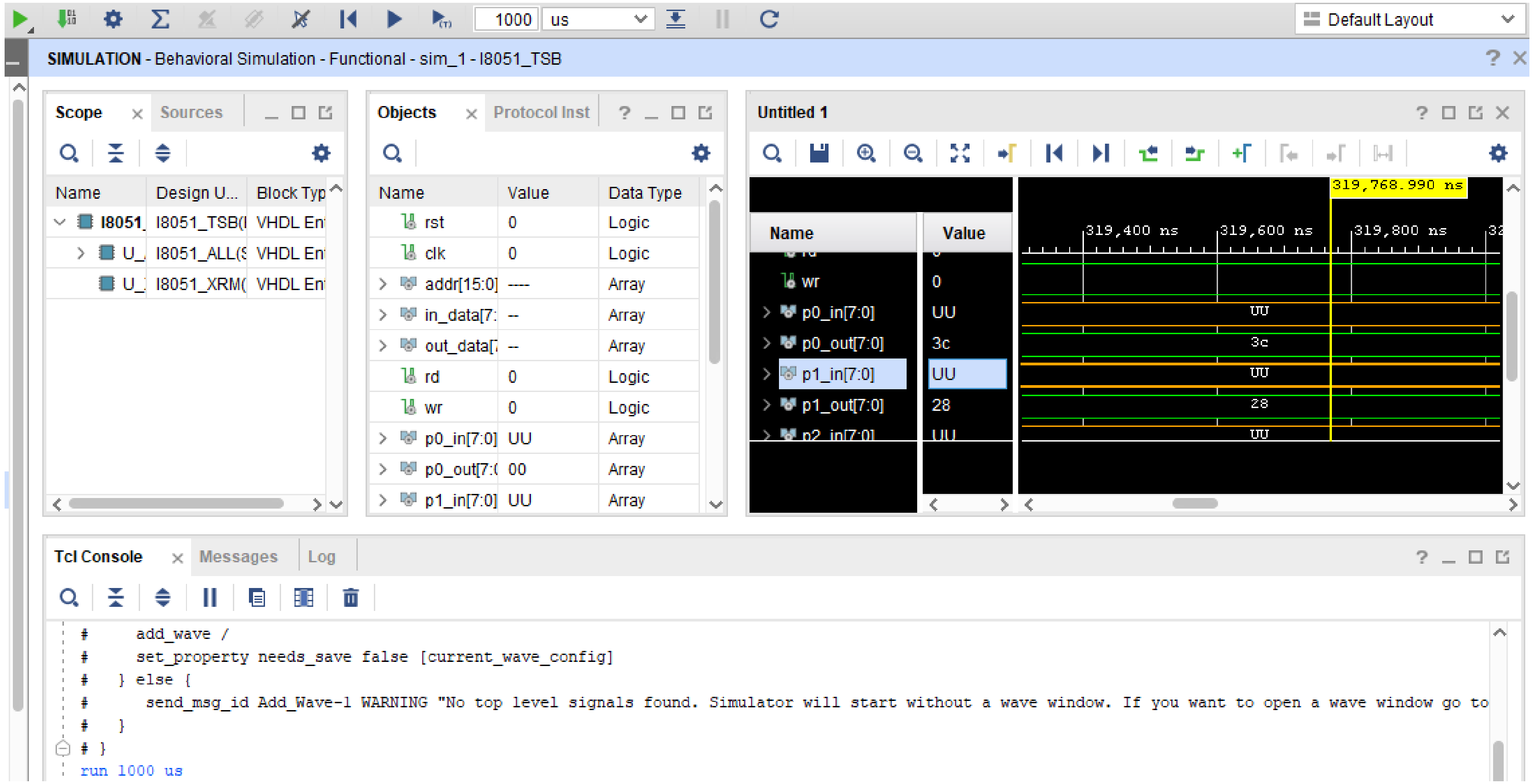
ocontroller/workspaces/euclid/i8051VHDL/i8051SDCCRefined/model/i8051_rom.vhd

```
482             "10000011",
483             "11110101", -- MOV_8
484             "10000011",
485             "00100010", -- RET
486             "01110101", -- MOV_12
487             "10000010",
488             "00000000",
489             "00100010");     -- RET
490         begin
491 
492             process(rst, clk)
493             begin
494                 if( rst = '1' ) then
```

```
run 1000 us
ERROR: Index 468 out of bound 0 to 467
Time: 4025 ns  Iteration: 0  Process: /I8051_TSB/U_ALL/U_ROM/line__492
  File: C:/Users/Agnese/Desktop/HOMELABS/1-Intel8051Microcontroller/workspaces/euclid/i8051VHDL/i8051SDCCRefined/model/i8051_rom.vhd

HDL Line: C:/Users/Agnese/Desktop/HOMELABS/1-Intel8051Microcontroller/workspaces/euclid/i8051VHDL/i8051SDCCRefined/model/i8051_rom.vhd:501
```

VHDL KEIL SIMULATION

The file compiled by Keil works properly.



SIMULATION - Behavioral Simulation - Functional - sim_1 - I8051_TSB

Name	Design U...	Block Typ
I8051_	I8051_TSB(I	VHDL En
U_	I8051_ALL(S	VHDL En
U_	I8051_XRM(	VHDL En

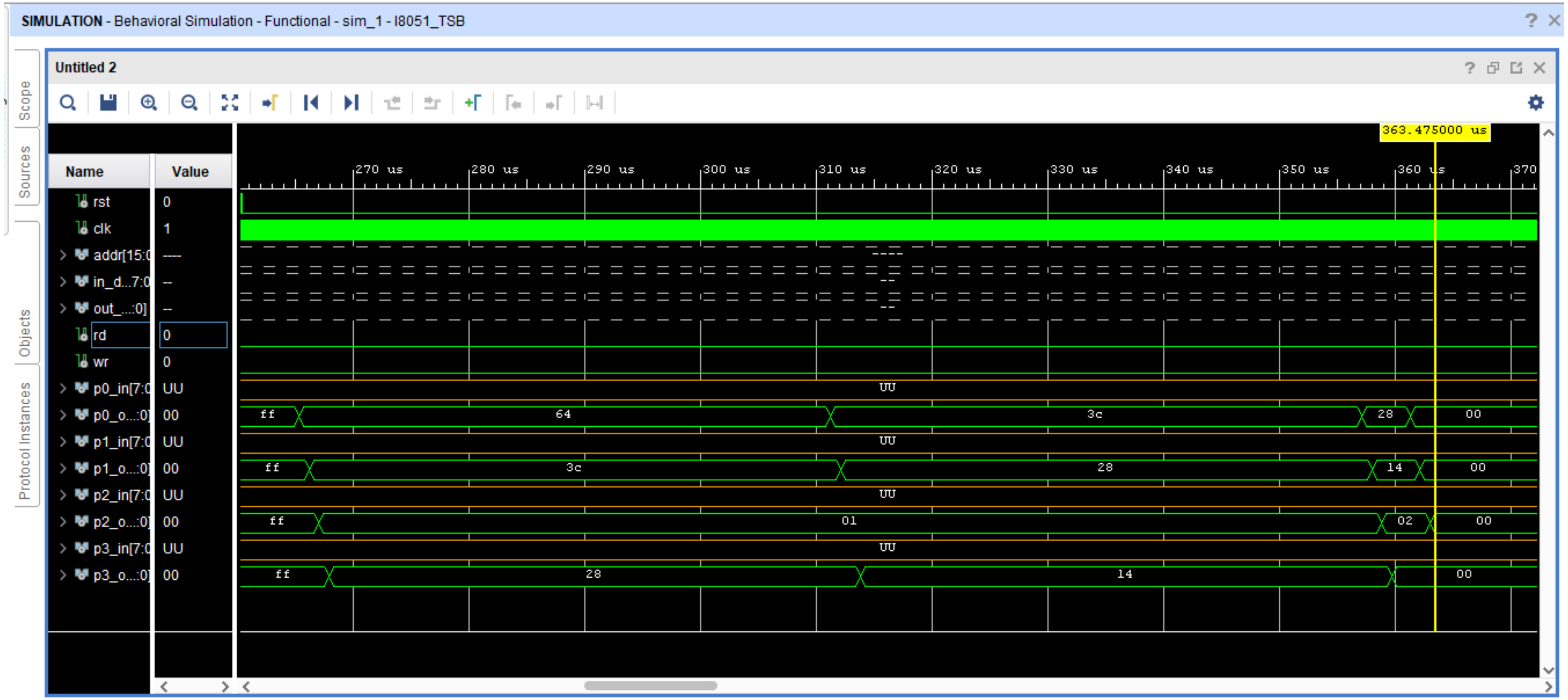
Name	Value	Data Type
rst	0	Logic
clk	0	Logic
addr[15:0]	----	Array
in_data[7:	--	Array
out_data[7	--	Array
rd	0	Logic
wr	0	Logic
p0_in[7:0]	UU	Array
p0_out[7:(	00	Array
p1_in[7:0]	UU	Array

Name	Value
wr	0
p0_in[7:0]	UU
p0_out[7:0]	3c
p1_in[7:0]	UU
p1_out[7:0]	28
p2_in[7:0]	UU

```
#     add_wave /
#     set_property needs_save false [current_wave_config]
#   } else {
#      send_msg_id Add_Wave-1 WARNING "No top level signals found. Simulator will start without a wave window. If you want to open a wave window go to
#   }
# }
run 1000 us
```

VHDL KEIL SIMULATION

VHDL Keil file simulation is in line with ISS one. Keil is to be the best compiler in this case too.



SIMULATION - Behavioral Simulation - Functional - sim_1 - I8051_TSB

Untitled 2

Name	Value
rst	0
clk	1
addr[15:0	----
in_d...7:0	--
out_...:0]	--
rd	0
wr	0
p0_in[7:0	UU
p0_o...:0]	00
p1_in[7:0	UU
p1_o...:0]	00
p2_in[7:0	UU
p2_o...:0]	00
p3_in[7:0	UU
p3_o...:0]	00

THANKS FOR YOUR ATTENTION