

# AndeSight

Advisor: Lih-Yih Chiou

Speaker: Elmer

Date: 2010/10/21



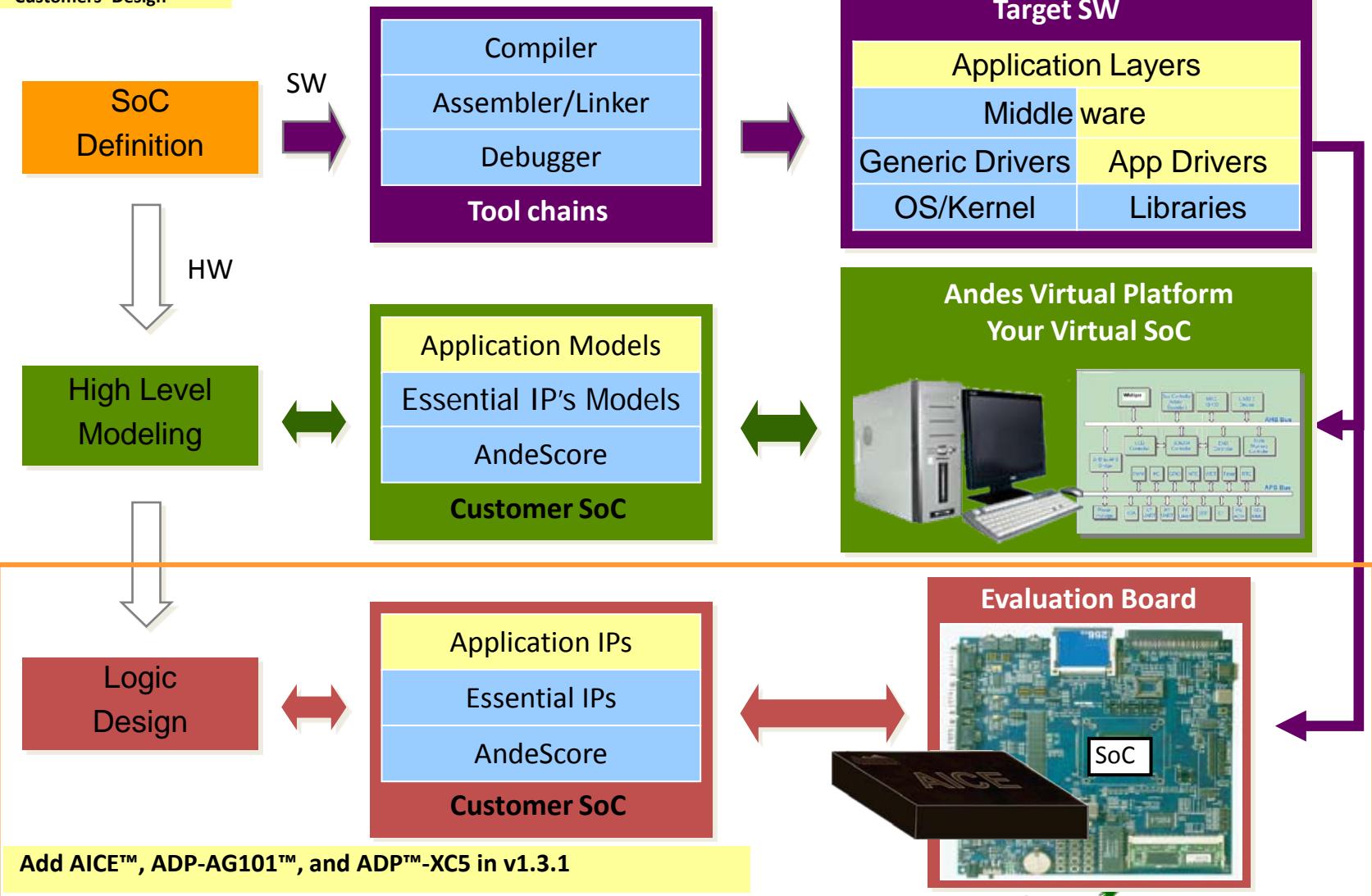
# Outline

- Introduction to AndeSight
- Start AndeSight
- Coder
- Debugger
- Profiler
- How to extract the useful machine code for our work?
- Reference

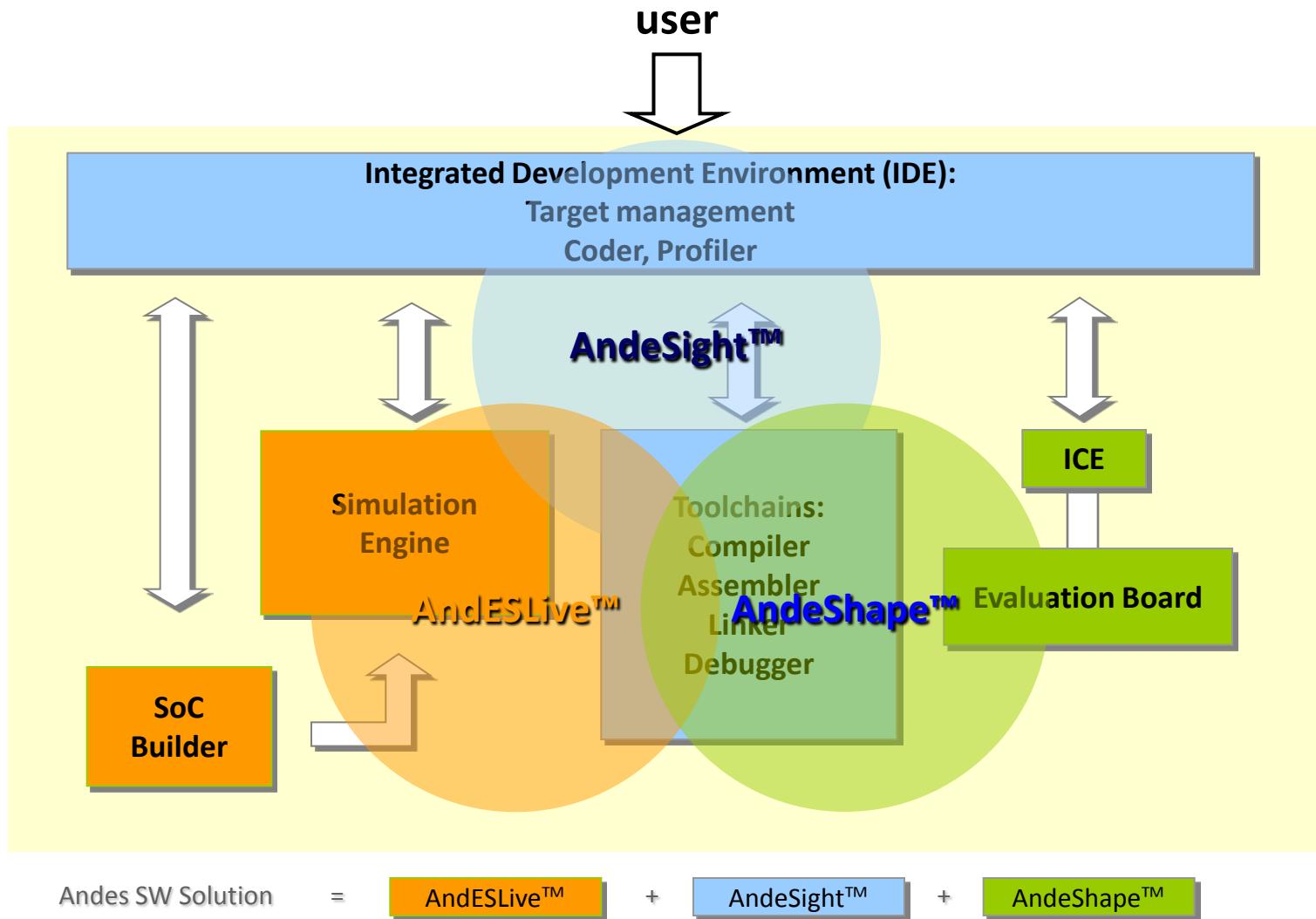
# Introduction to AndeSight(1/2)

Andes/Partners' solution

Customers' Design



# Introduction to AndeSight(2/2)



# Start AndeSight(1/2)



```
xterm
Platform = sparc64
Platform = sparc64
#####
# ModeSim run in the 32bit mode
#####
# If you want to run modelsim in the 64bit mode,
# do the next command in the command line.
#
# unix2 setenv MTI_WCO_MODE 64
#####
Platform = SOL2
#####
# 32BIT is the default mode
#
# If you want to run 64BIT mode,
# please set the LD_LIBRARY_PATH and SHLIB_PATH
# to path of 64BIT by yourself.
#####
#####
# Welcome to SoCLab-I !
# Available Workstations(Solaris): vlsicad1, vlsicad2, vlsicad3
# Tools: HSPICE, Laker, Verdi, SOCE, ModelSim, verilogXL, Calibre,
# DesignVision
# Available Workstations(Linux): vlsicad6
# Tools: MathLab, Colware2005, verilogXL, gcc/g++
#
# Quota: 100 MB for each account!
# When leaving, LOGOUT and SHUTDOWN the computer please!
# If there r any problem, contact Room 95316 please, thx!
# Email: soclab01@pvlsi.ee.ncku.edu.tw
#####
vlsicad1:/home/user1/efskyoo% ssh -X vlsicad4
efskyoo@vlsicad4's password: [REDACTED]
```



```
xterm
#####
# Welcome to SoCLab-I !
# Available Workstations(Solaris): vlsicad1, vlsicad2, vlsicad3
# Tools: HSPICE, Laker, Verdi, SOCE, ModelSim, verilogXL, Calibre,
# DesignVision
# Available Workstations(Linux): vlsicad6
# Tools: MathLab, Colware2005, verilogXL, gcc/g++
#
# Quota: 100 MB for each account!
# When leaving, LOGOUT and SHUTDOWN the computer please!
# If there r any problem, contact Room 95316 please, thx!
# Email: soclab01@pvlsi.ee.ncku.edu.tw
#####
vlsicad1:/home/user1/efskyoo% ssh -X vlsicad4
efskyoo@vlsicad4's password:
Last login: Thu Oct 7 15:27:45 2010 from vlsicad1
Platform = amd64
Platform = amd64
#####
# Welcome to SoCLab-I !
# Available Workstations(Solaris): vlsicad1, vlsicad2, vlsicad3
# Tools: HSPICE, Laker, Verdi, SOCE, ModelSim, verilogXL, Calibre,
# DesignVision
# Available Workstations(Linux): vlsicad6
# Tools: MathLab, Colware2005, verilogXL, gcc/g++
#
# Quota: 100 MB for each account!
# When leaving, LOGOUT and SHUTDOWN the computer please!
# If there r any problem, contact Room 95316 please, thx!
# Email: soclab01@pvlsi.ee.ncku.edu.tw
#####
vlsicad4:/home/user1/efskyoo%
```

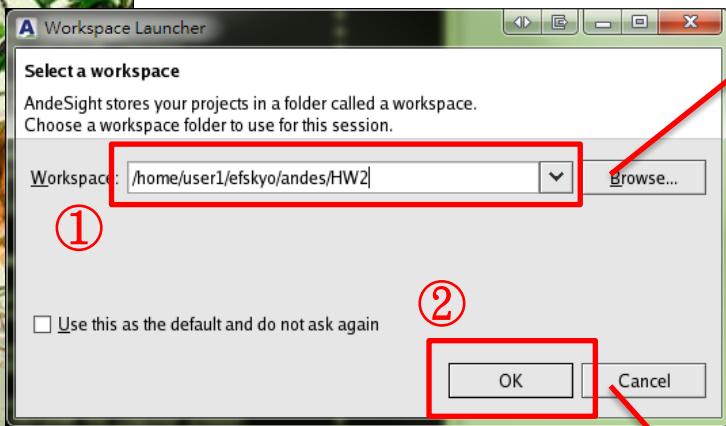


```
xterm
# Available Workstations(Linux): vlsicad6
# Tools: MathLab, Colware2005, verilogXL, gcc/g++
#
# Quota: 100 MB for each account!
# When leaving, LOGOUT and SHUTDOWN the computer please!
# If there r any problem, contact Room 95316 please, thx!
# Email: soclab01@pvlsi.ee.ncku.edu.tw
#####
vlsicad1:/home/user1/efskyoo% ssh -X vlsicad4
efskyoo@vlsicad4's password:
Last login: Fri Oct 8 12:48:48 2010 from vlsicad1
Platform = amd64
Platform = amd64
#####
# Welcome to SoCLab-I !
# Available Workstations(Solaris): vlsicad1, vlsicad2, vlsicad3
# Tools: HSPICE, Laker, Verdi, SOCE, ModelSim, verilogXL, Calibre,
# DesignVision
# Available Workstations(Linux): vlsicad6
# Tools: MathLab, Colware2005, verilogXL, gcc/g++
#
# Quota: 100 MB for each account!
# When leaving, LOGOUT and SHUTDOWN the computer please!
# If there r any problem, contact Room 95316 please, thx!
# Email: soclab01@pvlsi.ee.ncku.edu.tw
#####
vlsicad4:/home/user1/efskyoo% cd /usr/cad/AndeSight/ide/
```

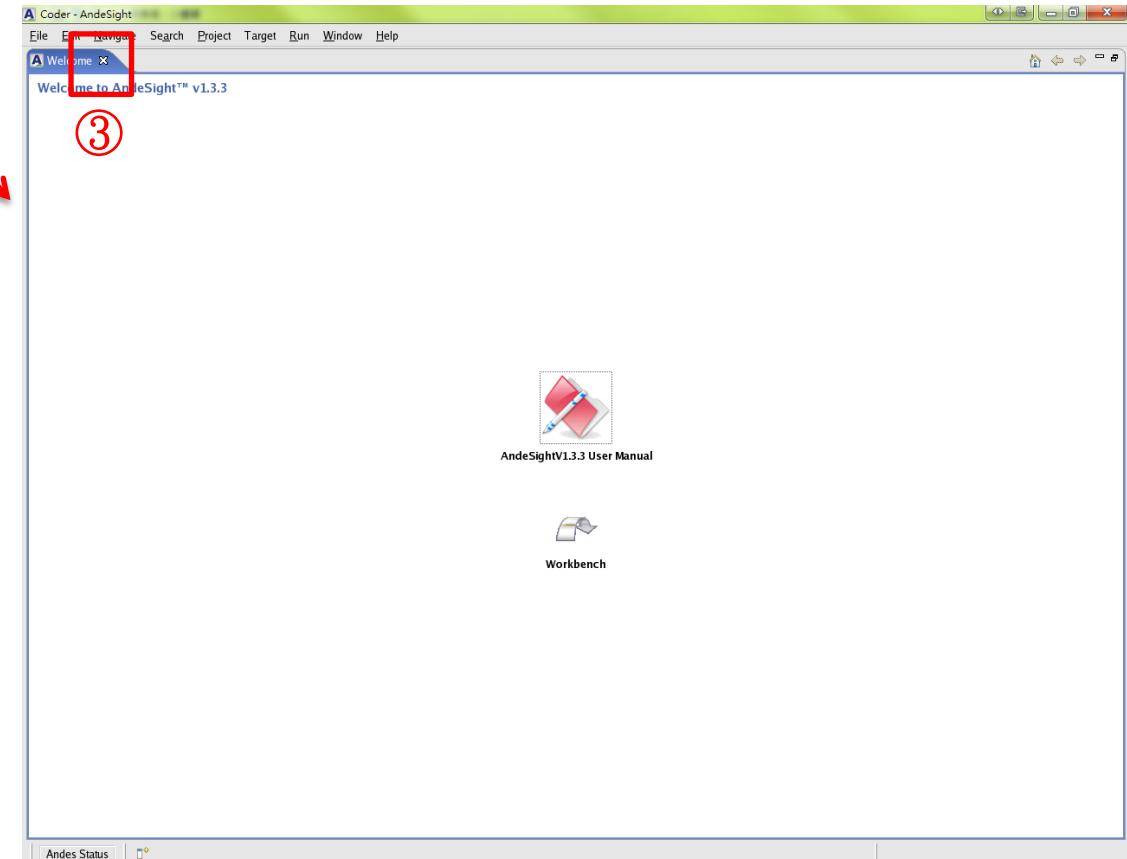


```
xterm
# Tools: HSPICE, Laker, Verdi, SOCE, ModelSim, verilogXL, Calibre,
# DesignVision
# Available Workstations(Linux): vlsicad6
# Tools: MathLab, Colware2005, verilogXL, gcc/g++
#
# Quota: 100 MB for each account!
# When leaving, LOGOUT and SHUTDOWN the computer please!
# If there r any problem, contact Room 95316 please, thx!
# Email: soclab01@pvlsi.ee.ncku.edu.tw
#####
vlsicad1:/home/user1/efskyoo% ssh -X vlsicad4
efskyoo@vlsicad4's password:
Last login: Fri Oct 8 13:33:48 2010 from vlsicad1
Platform = amd64
Platform = amd64
#####
# Welcome to SoCLab-I !
# Available Workstations(Solaris): vlsicad1, vlsicad2, vlsicad3
# Tools: HSPICE, Laker, Verdi, SOCE, ModelSim, verilogXL, Calibre,
# DesignVision
# Available Workstations(Linux): vlsicad6
# Tools: MathLab, Colware2005, verilogXL, gcc/g++
#
# Quota: 100 MB for each account!
# When leaving, LOGOUT and SHUTDOWN the computer please!
# If there r any problem, contact Room 95316 please, thx!
# Email: soclab01@pvlsi.ee.ncku.edu.tw
#####
vlsicad4:/home/user1/efskyoo% cd /usr/cad/AndeSight/ide/
vlsicad4:/usr/cad/AndeSight/ide/ ./start [REDACTED]
```

# Start AndeSight(2/2)

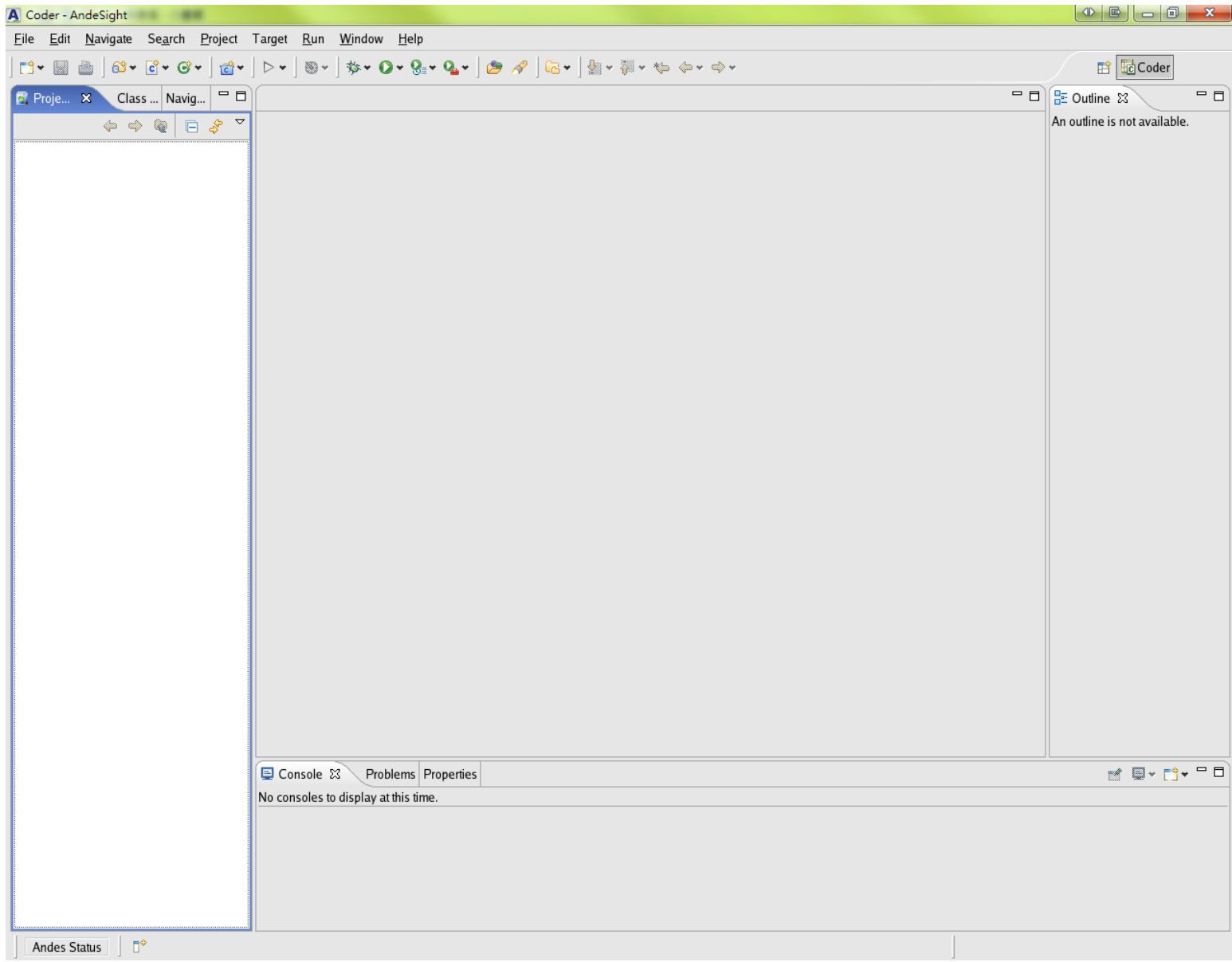


Please select your own directory!

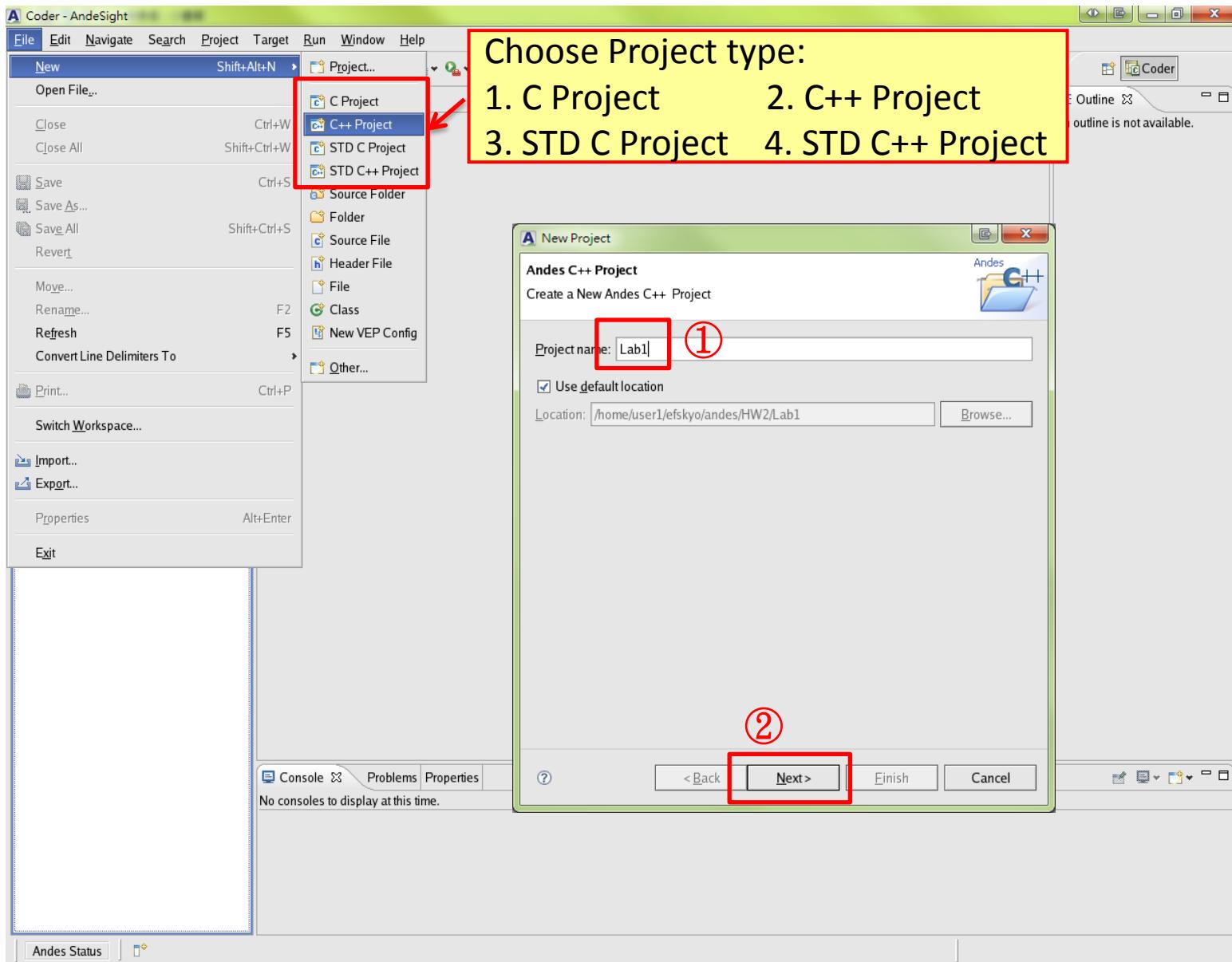


# Coder

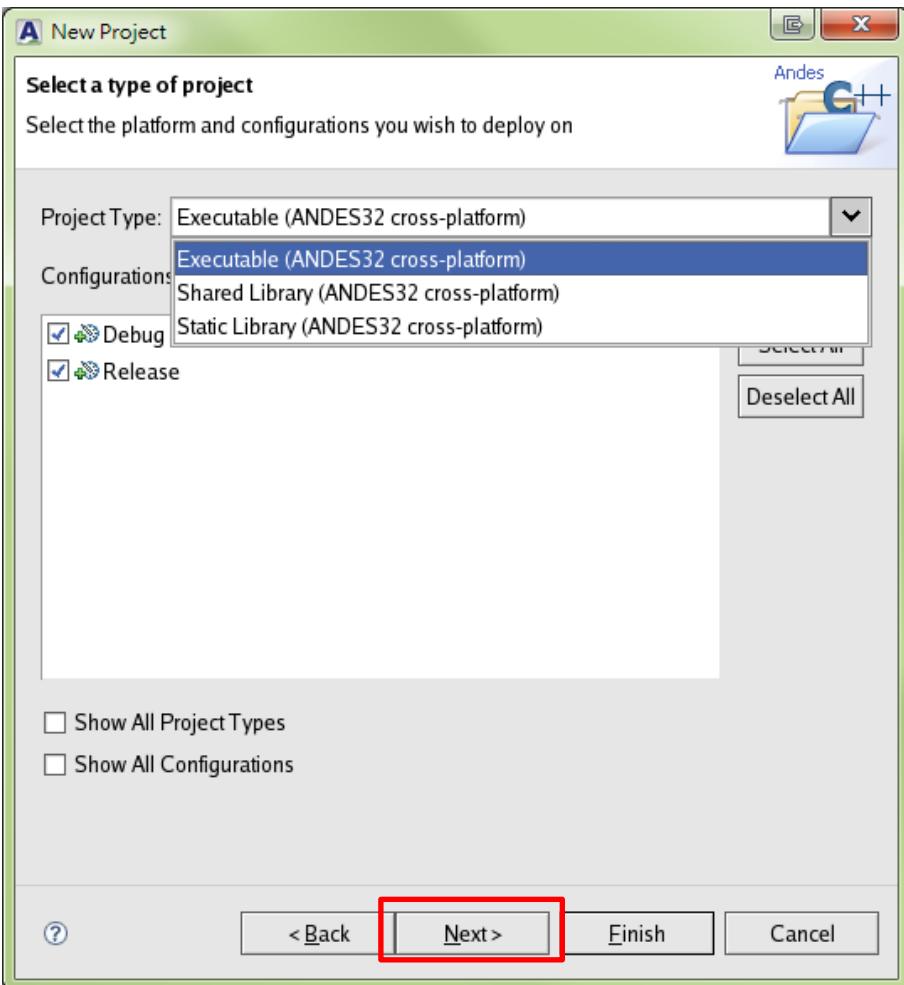
# Coder



# Coder – new project(1/3)



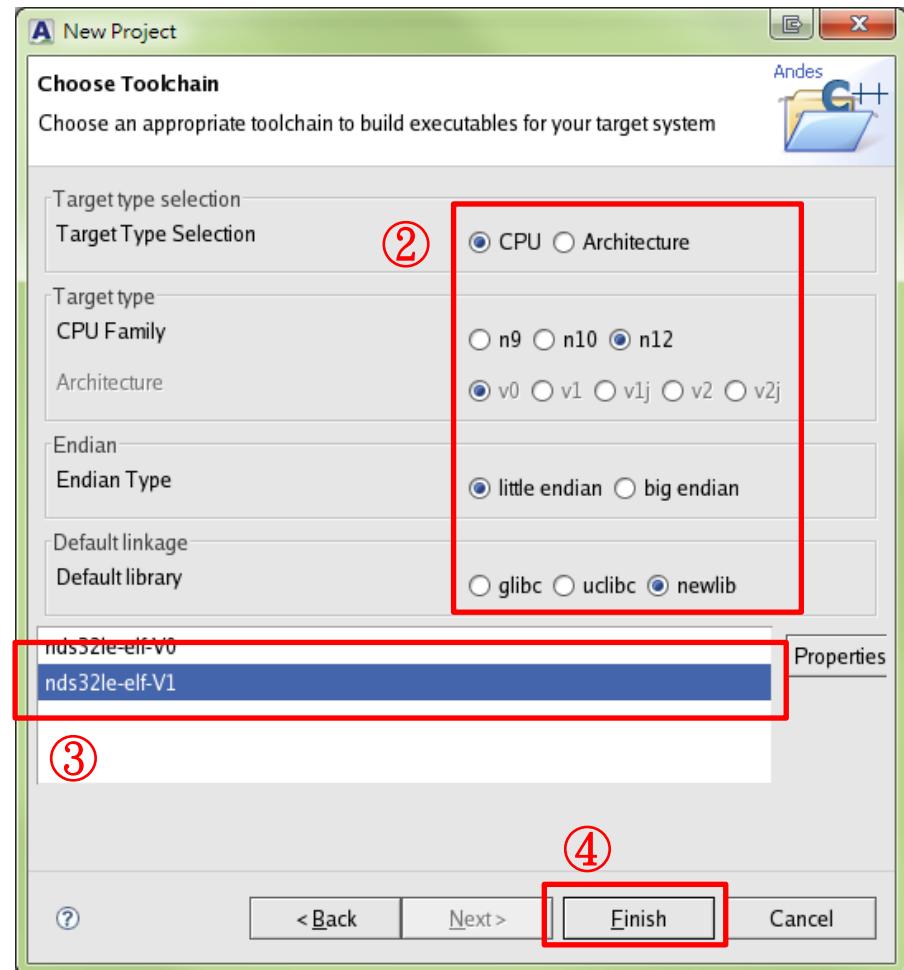
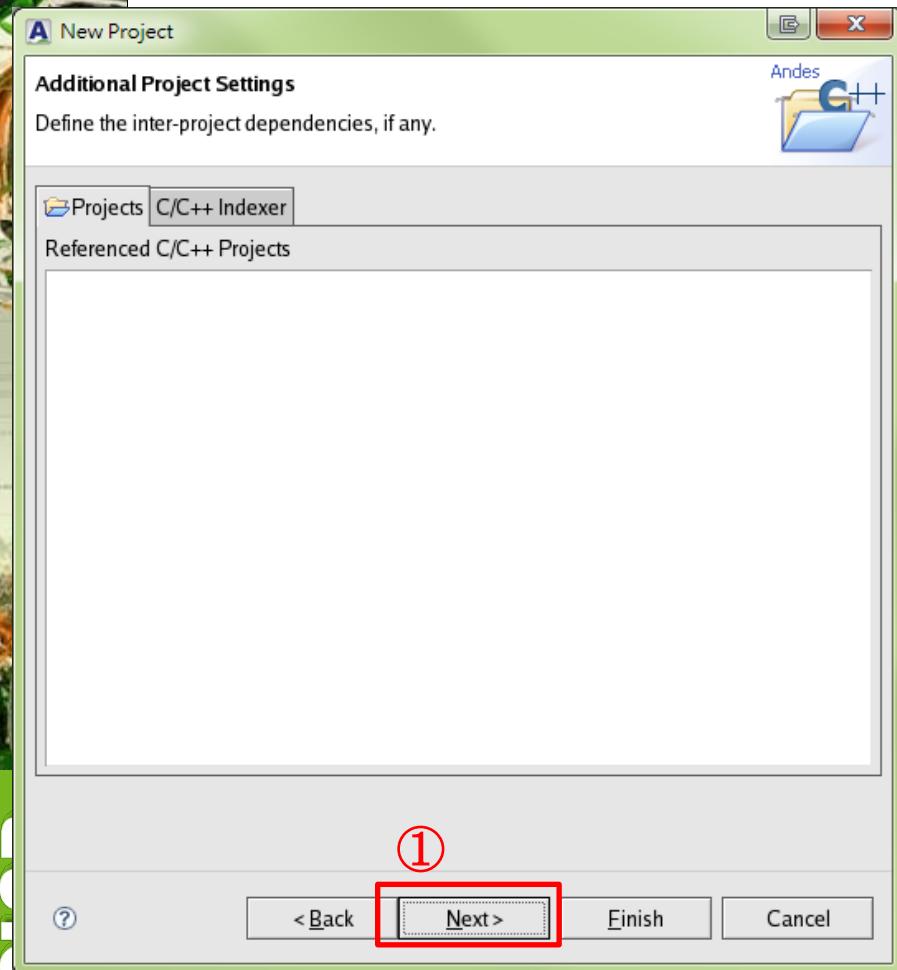
# Coder – new project(2/3)



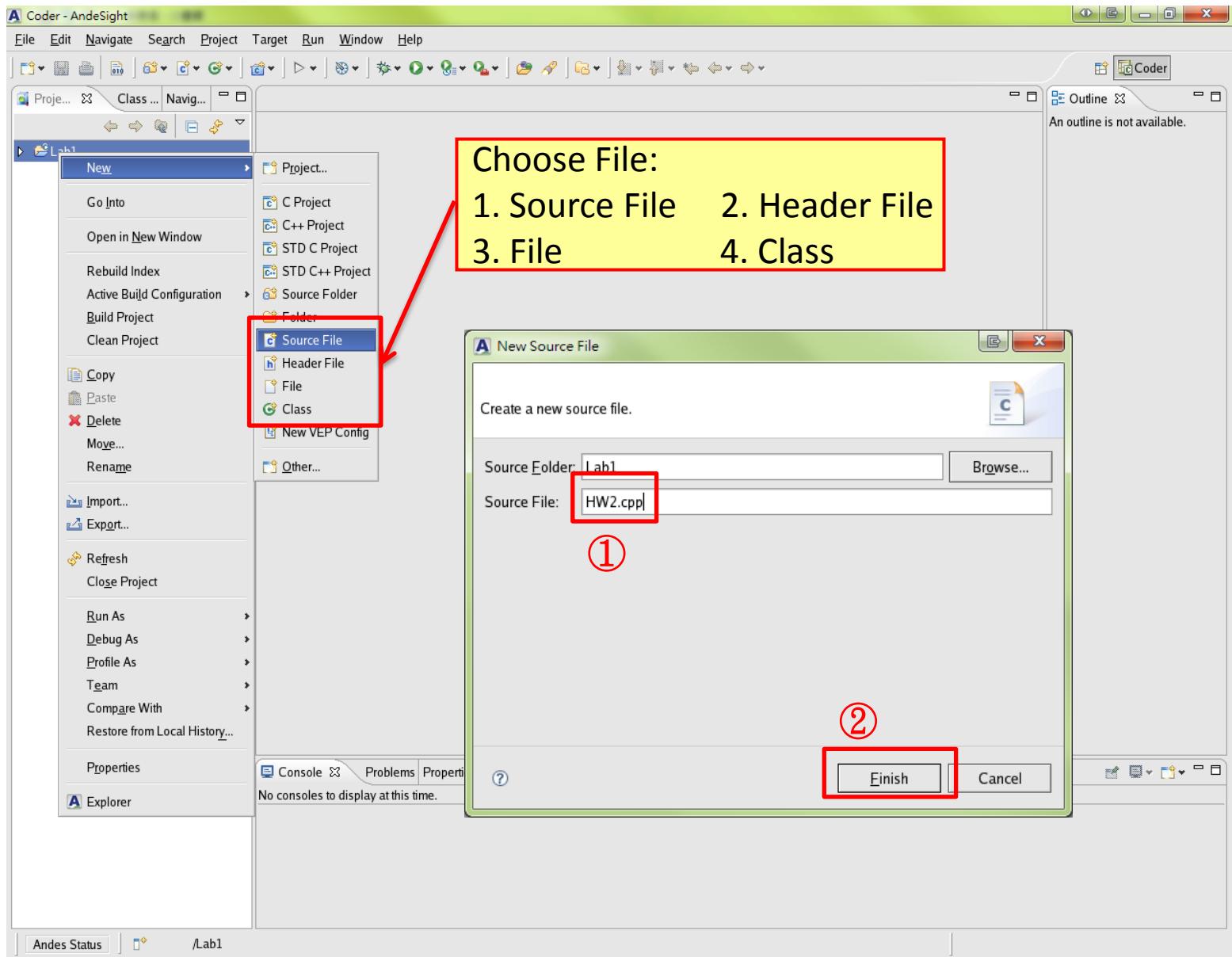
## Project Type:

1. Executable(ANDES32cross-platform)
2. Shared Library(ANDES32cross-platform)
3. Static Library(ANDES32cross-platform)

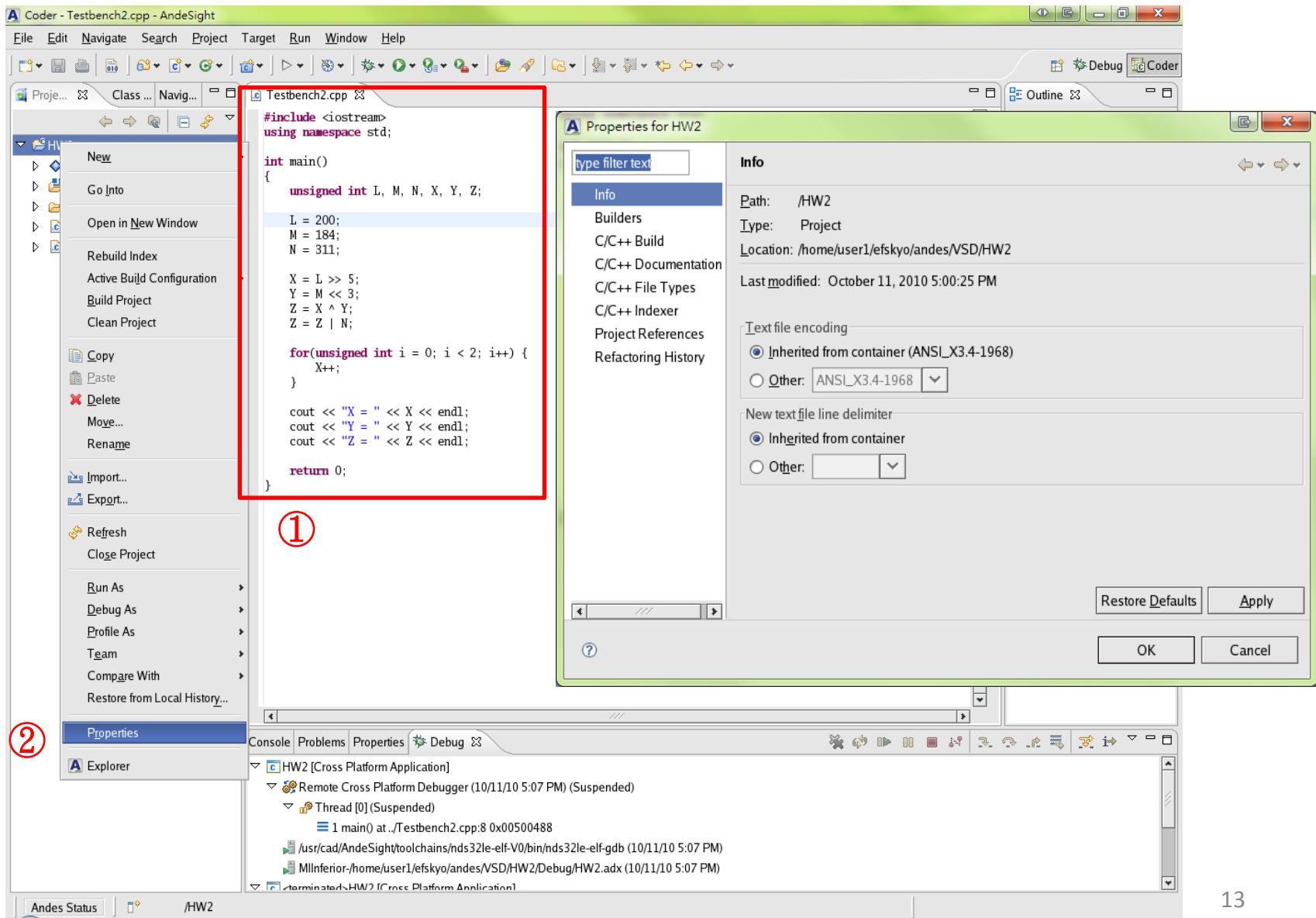
# Coder – new project(3/3)



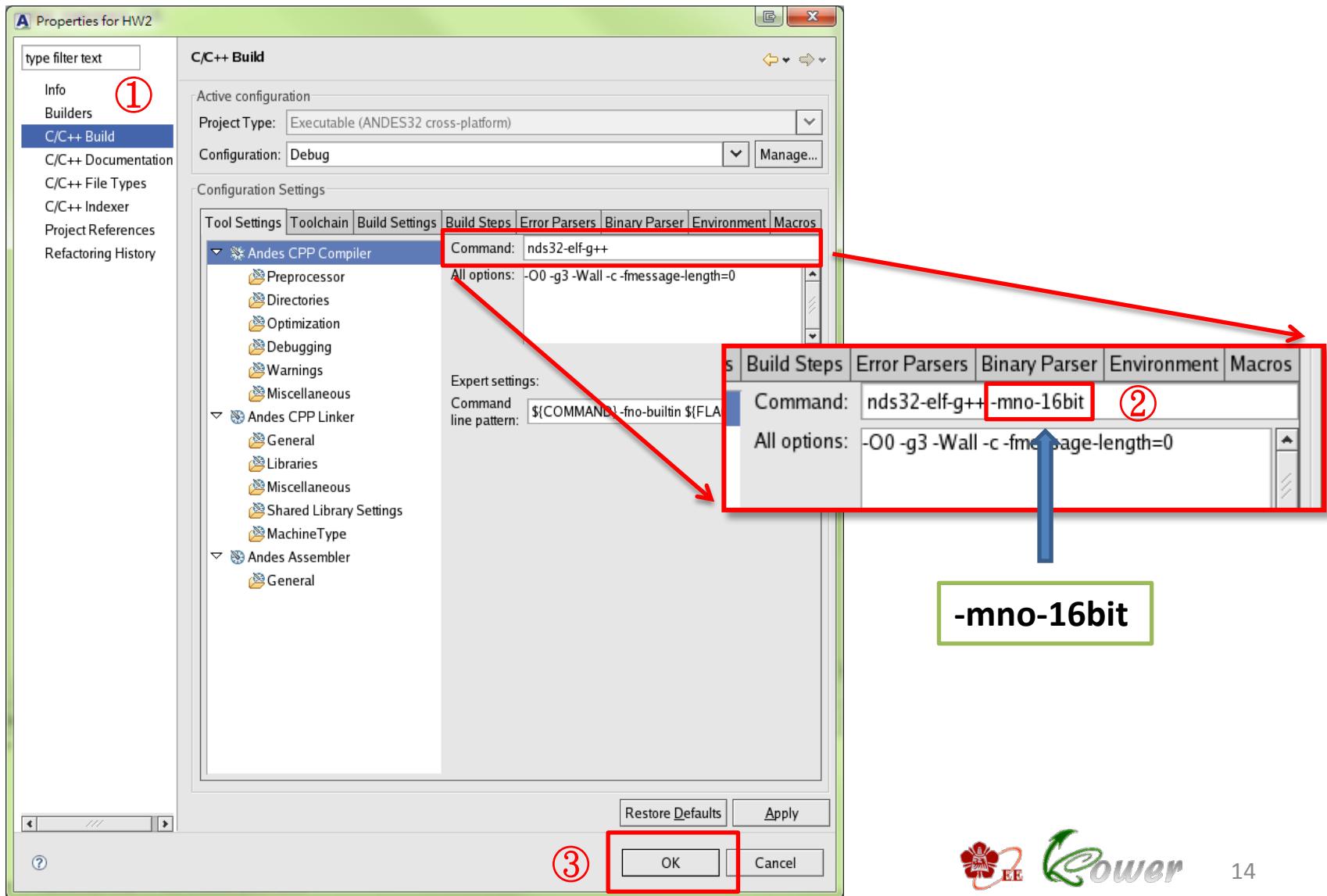
# Coder – new file



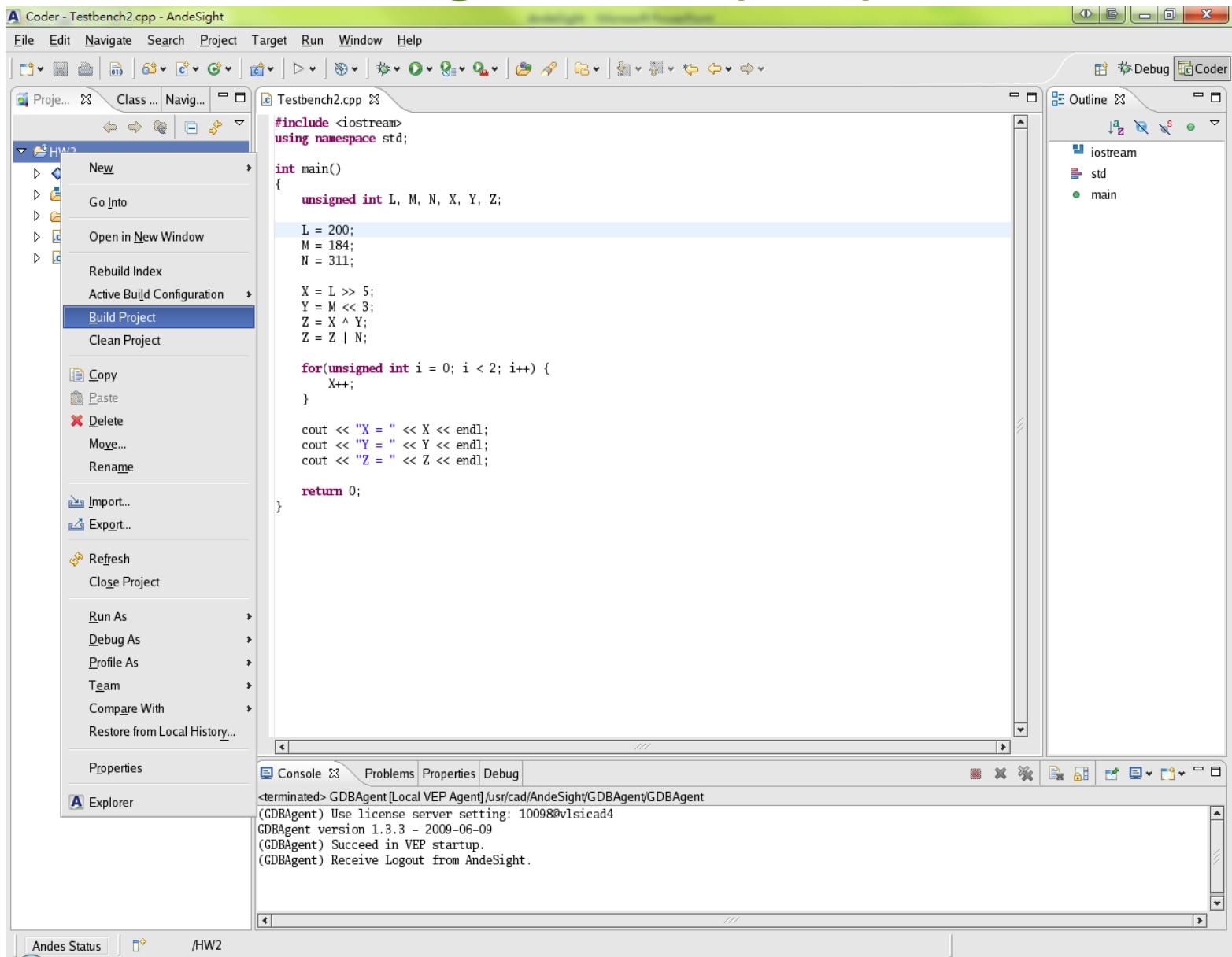
# Coder – coding & build(1/4)



# Coder – coding & build(2/4)



# Coder – coding & build(3/4)



# Coder – coding & build(4/4)

A Coder - HW2.cpp - AndeSight

File Edit Navigate Search Project Target Run Window Help

Proj... Class... Navig...

HW2.cpp

```
#include <iostream>
using namespace std;

int main()
{
    int a, b, c;

    a = 10;
    b = 20;

    a = a + 5;
    b = b + 15;
    c = a + b;

    return 0;
}
```

iostream  
std  
main

Console Problems Properties

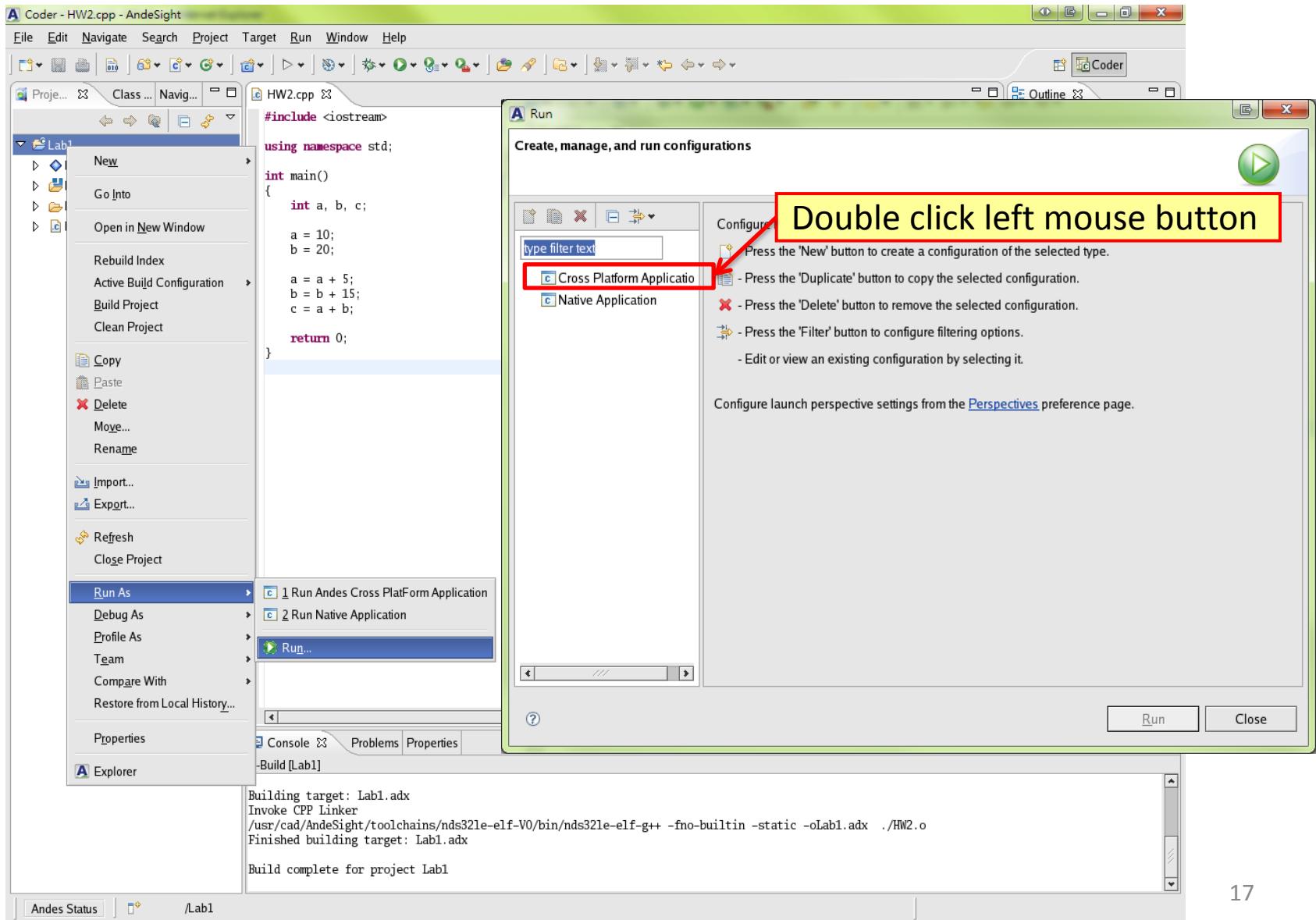
C-Build[Lab1]

```
Building target: Lab1.adx
Invoke CPP Linker
/usr/cad/AndeSight/toolchains/nds32le-elf-V0/bin/nds32le-elf-g++ -fno-builtin -static -oLab1.adx ./HW2.o
Finished building target: Lab1.adx

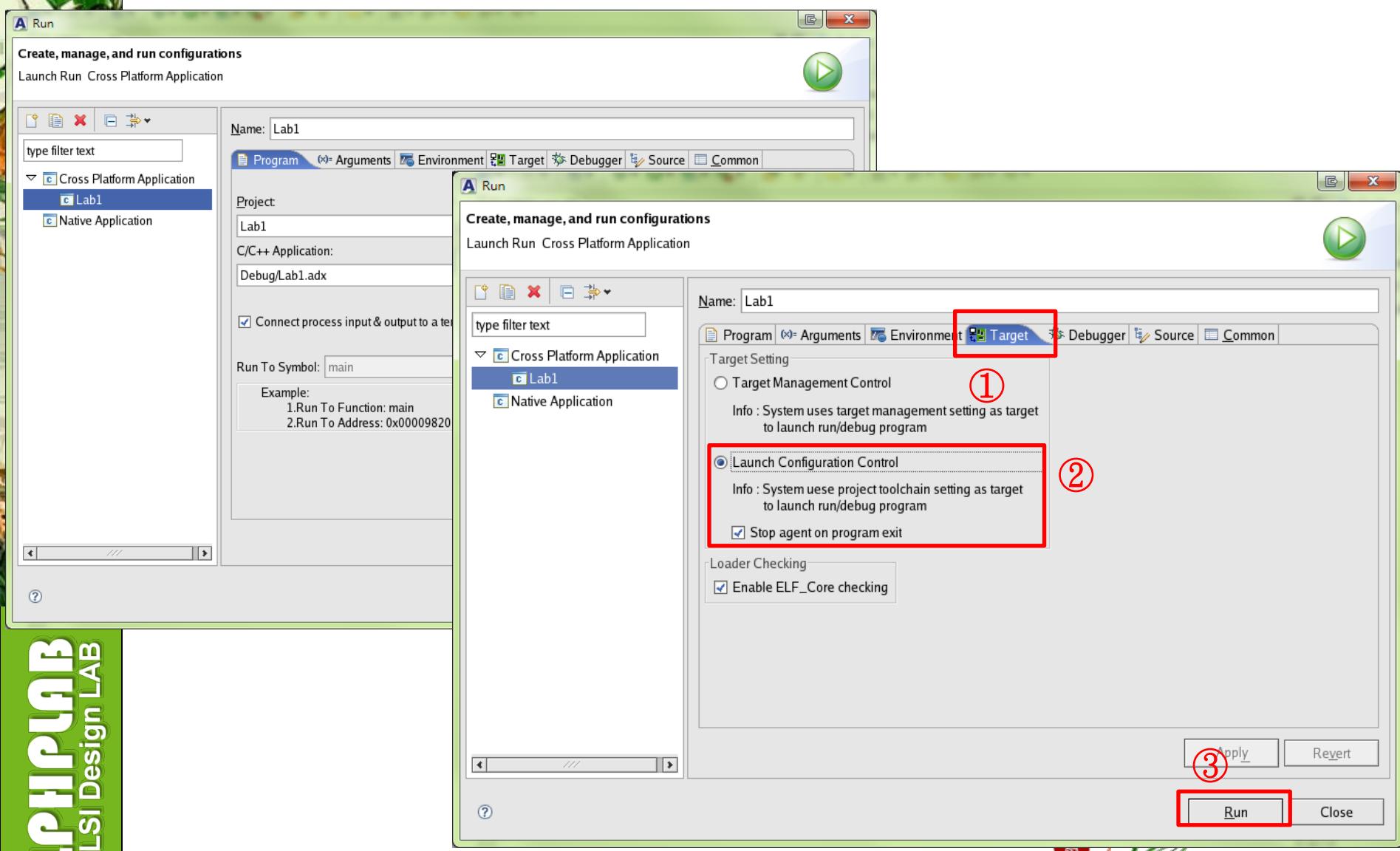
Build complete for project Lab1
```

Andes Status /Lab1

# Coder – run(1/5)



# Coder – run(2/5)



# Coder – run(3/5)

A Coder - HW2.cpp - AndeSight

File Edit Navigate Search Project Target Run Window Help

Proj... Class... Navig...

HW2.cpp x

```
#include <iostream>
using namespace std;
int main()
{
    int a, b, c;
    a = 10;
    b = 20;
    a = a + 5;
    b = b + 15;
    c = a + b;
    return 0;
}
```

Lab1

- Binaries
- Includes
- Debug
- HW2.cpp

Outline x

- iostream
- std
- main

Console x Problems Properties

```
<terminated> GDBAgent [Local VEP Agent]/usr/cad/AndeSight/GDBAgent/GDBAgent
(GDBAgent) Use license server setting: 10098@vlsicad4
GDBAgent version 1.3.3 - 2009-06-09
(GDBAgent) Succeed in VEP startup.
(GDBAgent) Receive Logout from AndeSight.
```

Andes Status Writable Smart Insert 18 : 1

# Coder – run(4/5)

A Coder - HW2\_2.cpp - AndeSight

File Edit Navigate Search Project Target Run Window Help

Proj... Class... Navig... HW2\_2.cpp

Lab1

#include <iostream>  
using namespace std;

int main()  
{  
 unsigned int L, M, N, X, Y, Z;  
  
 L = 200;  
 M = 184;  
 N = 311;  
  
 X = L >> 5;  
 Y = M << 3;  
 Z = X ^ Y;  
 Z = Z | N;  
  
 cout << "X = "  
 cout << "Y = "  
 cout << "Z = "  
}  
return 0;

1 C-Build [Lab1]  
2 Lab1 [Cross Platform Application] /usr/cad/AndeSight/toolchains/nds32le-elf-V0/bin/nds32le-elf-gdb (10/9/10 3:31 PM)  
3 Lab1 [Cross Platform Application] Mlinferior-/home/user1/efskyo/andes/HW2/Lab1/Debug/Lab1.adx (10/9/10 3:31 PM)  
4 Lab1 [Cross Platform Application] /usr/cad/AndeSight/toolchains/nds32le-elf-V0/bin/nds32le-elf-gdb (10/9/10 3:36 PM)  
5 Lab1 [Cross Platform Application] Mlinferior-/home/user1/efskyo/andes/HW2/Lab1/Debug/Lab1.adx (10/9/10 3:36 PM)  
6 Lab1 [Cross Platform Application] /usr/cad/AndeSight/toolchains/nds32le-elf-V0/bin/nds32le-elf-gdb (10/9/10 3:40 PM)  
7 Lab1 [Cross Platform Application] Mlinferior-/home/user1/efskyo/andes/HW2/Lab1/Debug/Lab1.adx (10/9/10 3:40 PM)  
8 Lab1 [Cross Platform Application] /usr/cad/AndeSight/toolchains/nds32le-elf-V0/bin/nds32le-elf-gdb (10/9/10 3:46 PM)  
9 Lab1 [Cross Platform Application] Mlinferior-/home/user1/efskyo/andes/HW2/Lab1/Debug/Lab1.adx (10/9/10 3:46 PM)  
Lab1 [Cross Platform Application] /usr/cad/AndeSight/toolchains/nds32le-elf-V0/bin/nds32le-elf-gdb (10/10/10 9:10 AM)  
Lab1 [Cross Platform Application] Mlinferior-/home/user1/efskyo/andes/HW2/Lab1/Debug/Lab1.adx (10/10/10 9:10 AM)  
Lab1 [Cross Platform Application] /usr/cad/AndeSight/toolchains/nds32le-elf-V0/bin/nds32le-elf-gdb (10/10/10 10:52 AM)  
Lab1 [Cross Platform Application] Mlinferior-/home/user1/efskyo/andes/HW2/Lab1/Debug/Lab1.adx (10/10/10 10:52 AM)  
Lab1 [Cross Platform Application] /usr/cad/AndeSight/toolchains/nds32le-elf-V0/bin/nds32le-elf-gdb (10/10/10 10:56 AM)  
Lab1 [Cross Platform Application] Mlinferior-/home/user1/efskyo/andes/HW2/Lab1/Debug/Lab1.adx (10/10/10 10:56 AM)  
✓ <terminated> GDBAgent [Local VEP Agent] /usr/cad/AndeSight/GDBAgent/GDBAgent  
<terminated> Lab1 [Cross Platform Application] /usr/cad/AndeSight/toolchains/nds32le-elf-V0/bin/nds32le-elf-gdb (10/11/10 9:44 AM)  
<terminated> Lab1 [Cross Platform Application] Mlinferior-/home/user1/efskyo/andes/HW2/Lab1/Debug/Lab1.adx (10/11/10 9:44 AM)

Console Problem

<terminated> GDBAgent [Local VEP Agent] /usr/cad/AndeSight/GDBAgent/GDBAgent  
(GDBAgent) Use license server setting: 10098@vlsicad4  
GDBAgent version 1.3.3 - 2009-06-09  
(GDBAgent) Succeed in VEP startup.  
(GDBAgent) Receive Logout from AndeSight.

Andes Status

20

# Coder – run(5/5)

The screenshot shows the AndesIDE Coder interface. The main window displays a CPP file named HW2\_2.cpp. The code includes declarations for iostream and std, defines variables L, M, N, X, Y, and Z, and performs bit operations. A red box highlights the cout statements that output the values of X, Y, and Z. The output is also shown in the Console tab, where the values 6, 1472, and 1527 are displayed. The AndesVLSI logo is visible on the left.

```
#include <iostream>
using namespace std;

int main()
{
    unsigned int L, M, N, X, Y, Z;

    L = 200;
    M = 184;
    N = 311;

    X = L >> 5;
    Y = M << 3;
    Z = X ^ Y;
    Z = Z | N;

    cout << "X = " << X << endl;
    cout << "Y = " << Y << endl;
    cout << "Z = " << Z << endl;

    return 0;
}
```

<terminated>Lab1 [Cross Platform Application] MilnInferior-/home/user1/efskyo/andes/HW2/Lab1/Debug/Lab1.adx (10/11/10 9:44 AM)  
X = 6  
Y = 1472  
Z = 1527



# Debugger

# Debugger(1/3)

The screenshot shows the AndeSight IDE interface. On the left, there's a vertical toolbar with icons for file operations like Open, Save, and Print. Below that is a context menu for a selected item in the project tree, with options like 'New', 'Go Into', 'Open in New Window', 'Rebuild Index', 'Active Build Configuration', 'Build Project', 'Clean Project', 'Copy', 'Paste', 'Delete', 'Move...', 'Rename', 'Import...', 'Export...', 'Refresh', 'Close Project', 'Run As' (selected), 'Profile As', 'Team', 'Compare With', 'Restore from Local History...', 'Properties', and 'Explorer'. The 'Run As' submenu shows '1 Debug Andes Cross PlatForm Application', '2 Debug Native Application', and 'Debug...'. The main workspace contains a code editor with the following C++ code:

```
#include <iostream>

using namespace std;

int main()
{
    int a, b, c;

    a = 10;
    b = 20;

    a = a + 5;
    b = b + 15;
    c = a + b;

    return 0;
}
```

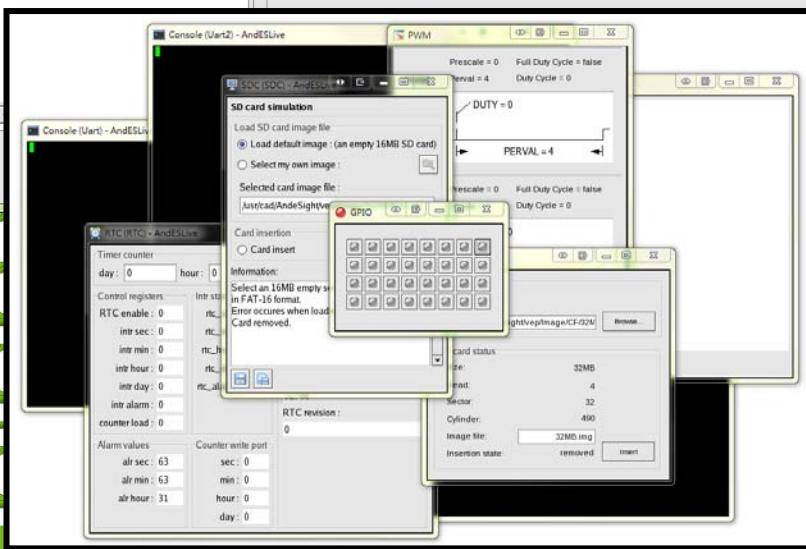
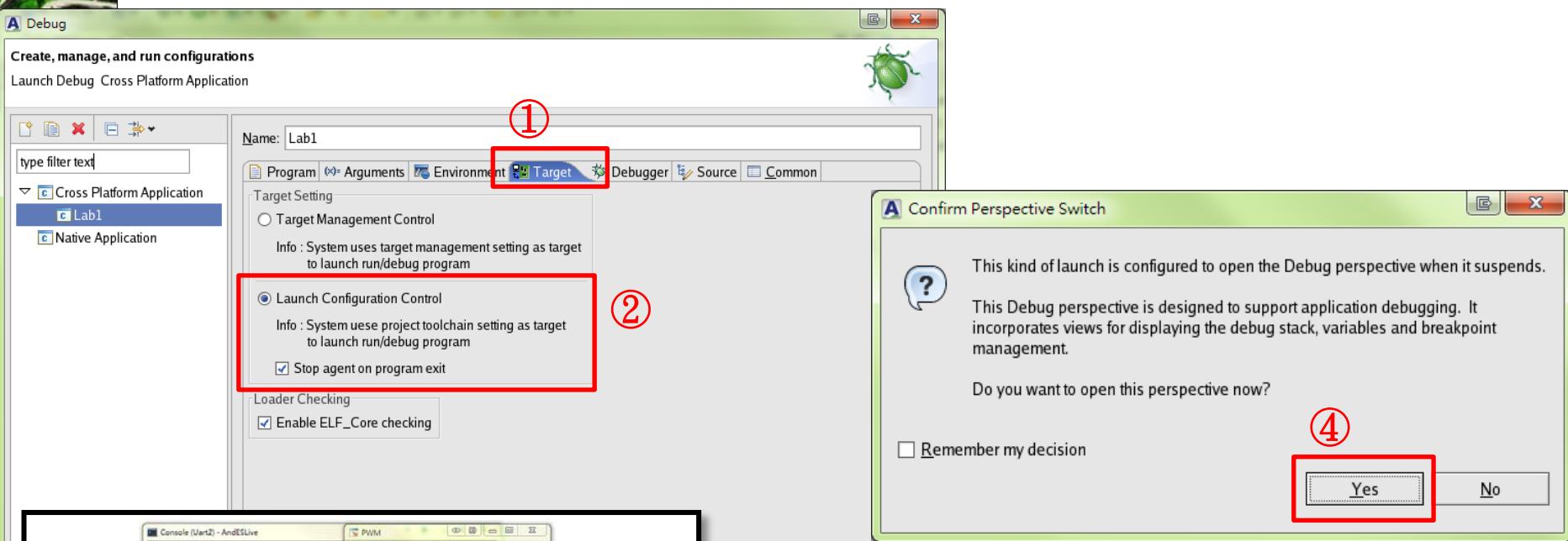
To the right of the code editor is a 'Debug' dialog box titled 'Create, manage, and run configurations'. It shows a list of configurations: 'Cross Platform Application' (selected) with 'Lab1' and 'Native Application' listed under it. The 'Program' tab is active, showing 'Name: Lab1', 'Project: Lab1', 'C/C++ Application: Debug/Lab1.adx', and a checked checkbox 'Connect process input & output to a terminal.' Below that is a 'Run To Symbol:' field with 'main' selected. At the bottom of the dialog are 'Apply', 'Revert', 'Debug', and 'Close' buttons.

At the bottom of the interface, there's a 'Console' tab showing the output of the GDBAgent startup:

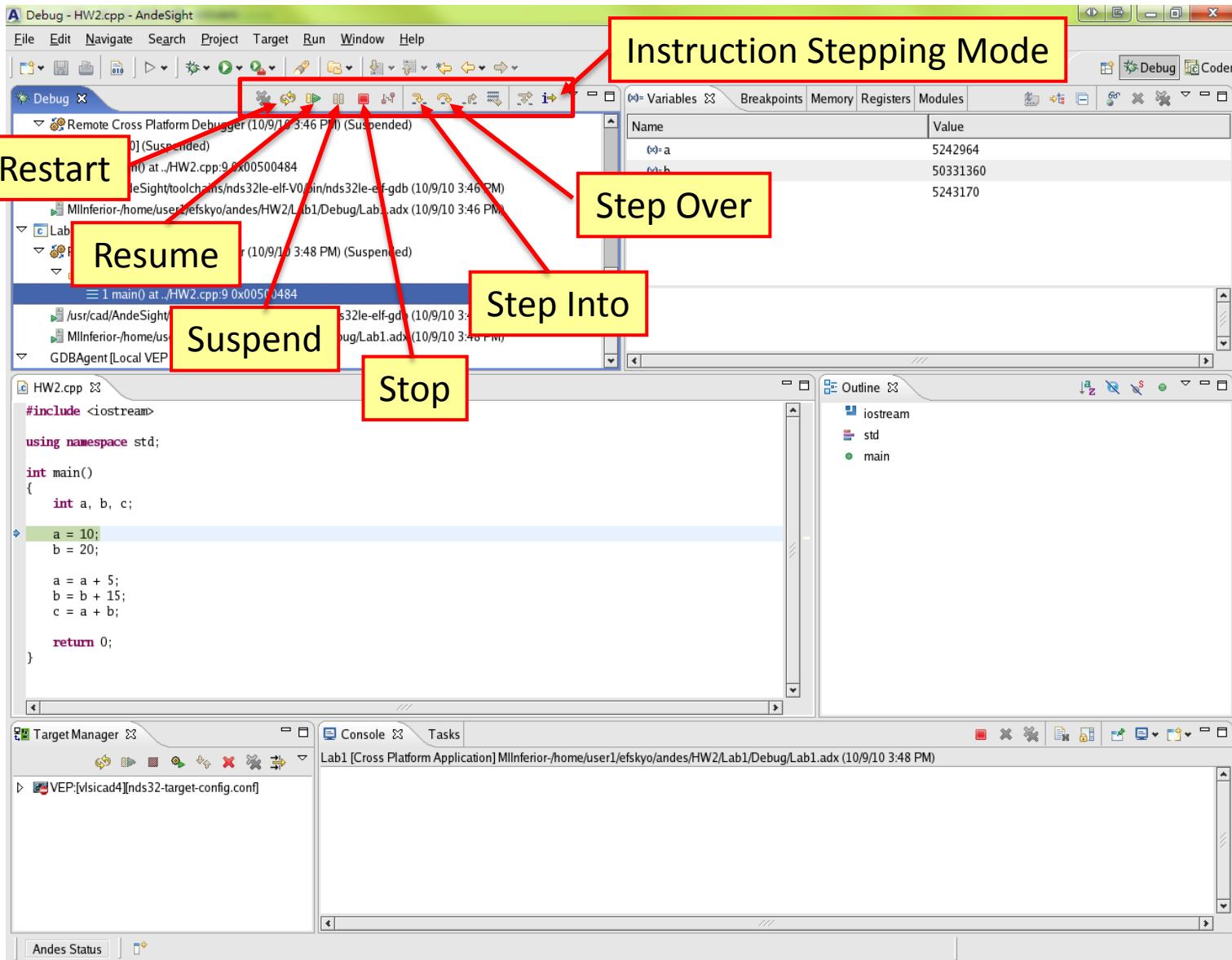
```
<terminated> GDBAgent [Local VEP Agent]/usr/cad/AndeSight/GDBAgent/GDBAgent
(GDBAgent) Use license server setting: 10098@vlsicad4
GDBAgent version 1.3.3 - 2009-06-09
(GDBAgent) Succeed in VEP startup.
(GDBAgent) Receive Logout from AndeSight.
```

On the far left, there's a vertical banner with the text 'UPHILAB VLSI Design LAB'.

# Debugger(2/3)



# Debugger(3/3)



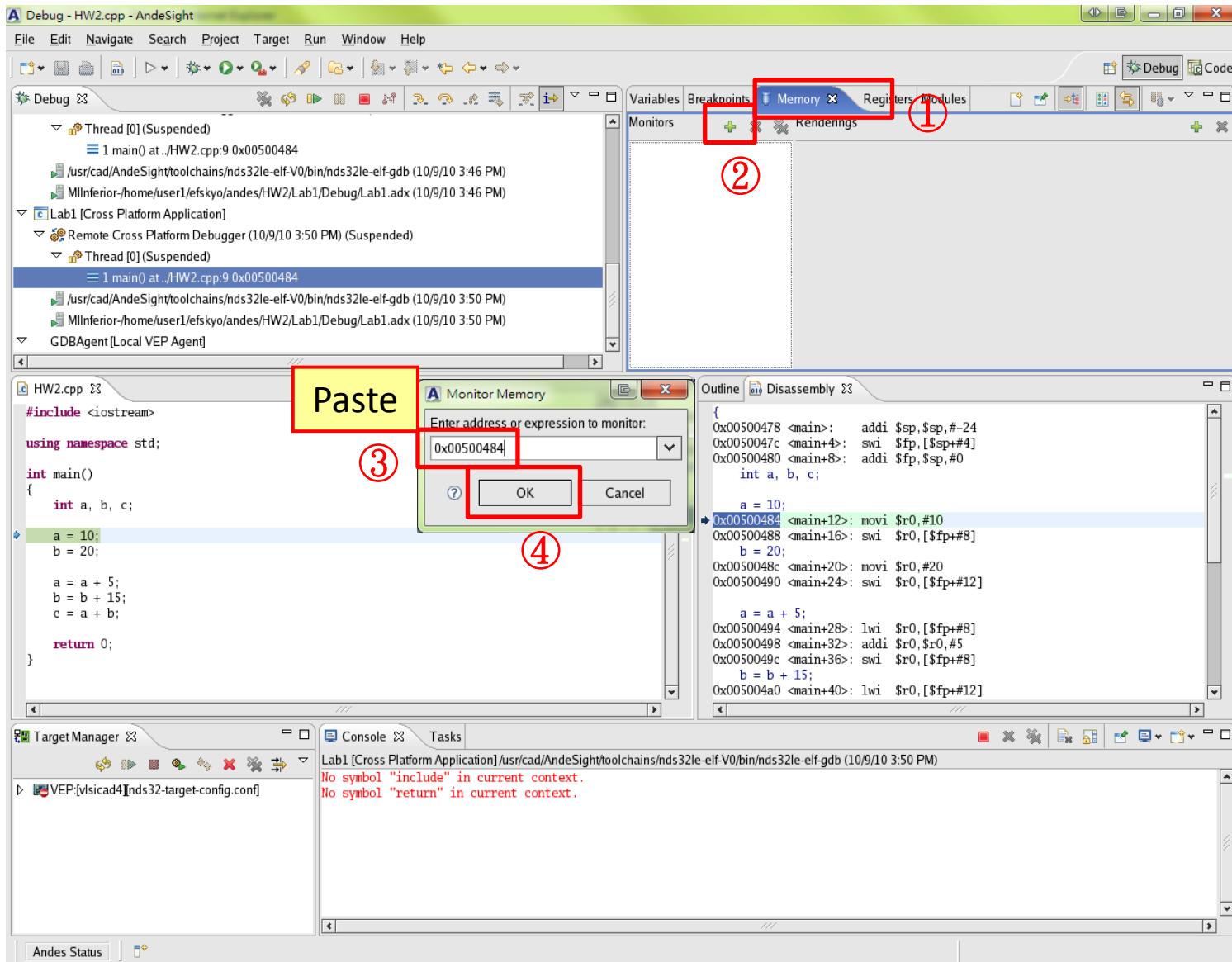
# Disassembly

The screenshot shows the AndeSight IDE interface with several windows open:

- Debug View (Top Left):** Shows the current state of threads. A red circle labeled ① points to the "Step Into" button in the toolbar.
- Variables View (Top Right):** Displays the values of variables `a`, `b`, and `c`.

Name	Value
<code>a</code>	5242964
<code>b</code>	50331360
<code>c</code>	5243170
- Code View (Bottom Left):** Shows the C++ source code for `HW2.cpp`. A yellow box labeled "Copy" highlights the line `a = 10;`.
- Disassembly View (Bottom Right):** Shows the assembly code corresponding to the C++ code. A red circle labeled ② points to the instruction `movi $r0,#10` at address `0x00500484`.
- Target Manager View (Bottom Left):** Shows the target configuration `nds32-target-config.conf`.
- Console View (Bottom Center):** Displays the output of the build process, indicating that no symbol "include" was found.

# Memory Address

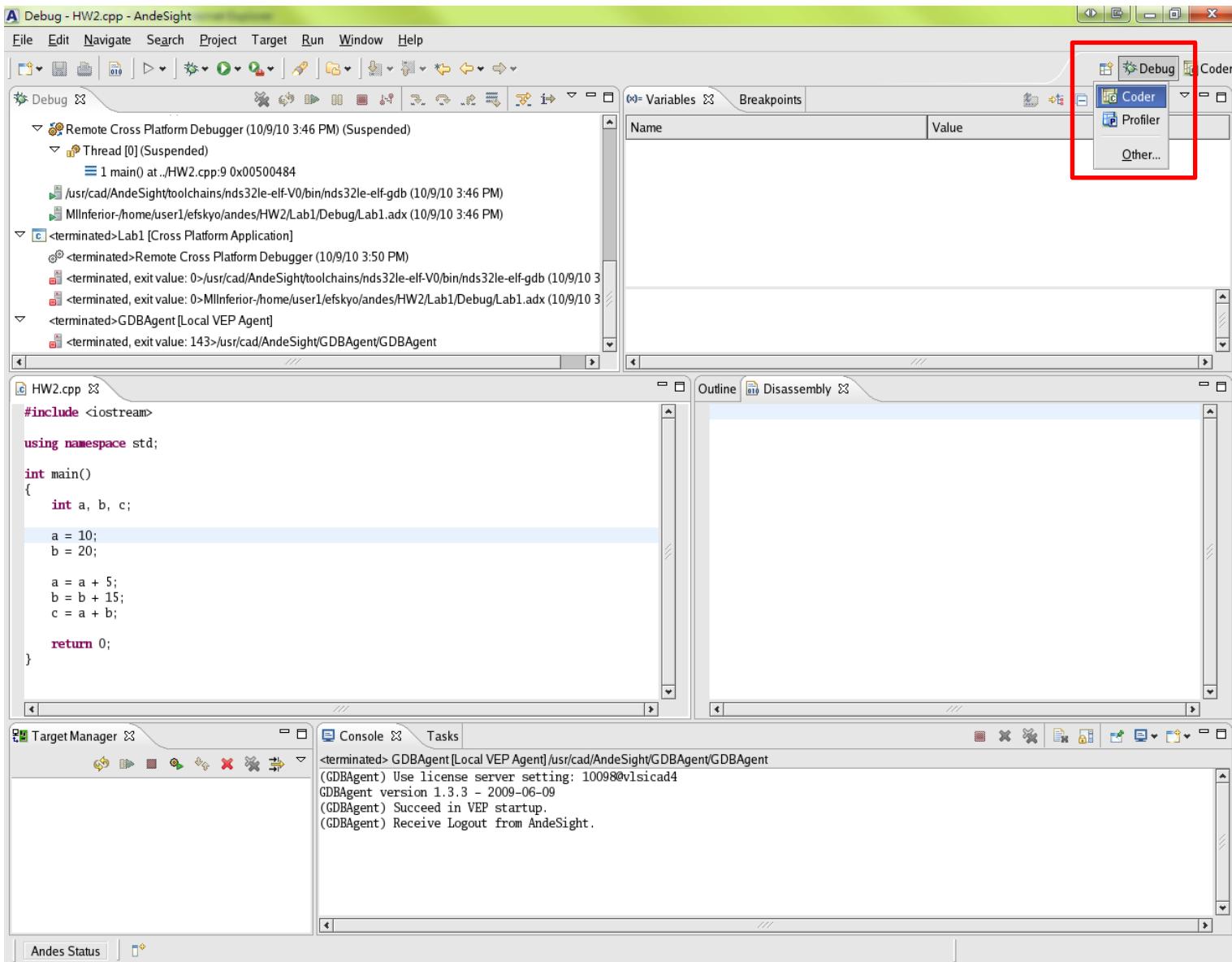


# Machine Code

The screenshot shows the AndeSight debugger interface with the following windows:

- Threads:** Shows two threads: Thread [0] (Suspended) at 0x00500484 and Lab1 [Cross Platform Application] (Suspended) at 0x00500484.
- Memory:** A hex dump of memory starting at address 0x00500484. The first few bytes are highlighted with a red box: 51CF8000 (440000A).
- Disassembly:** Shows the assembly code for the main() function. The instruction at address 0x00500484 is highlighted with a red box: movi \$r0,#10.
- Source Code:** The C++ source code for HW2.cpp is shown, with the main() function highlighted.
- Target Manager:** Shows the target configuration: VEP:[vlsicad4]nds32-target-config.conf.
- Console:** Displays the command: Lab1 [Cross Platform Application] /usr/cad/AndeSight/toolchains/nds32le-elf-V0/bin/nds32le-elf-gdb (10/9/10 3:50 PM). It also shows error messages: "No symbol "include" in current context.", "No symbol "return" in current context.", and "Attempt to use a type name as an expression".

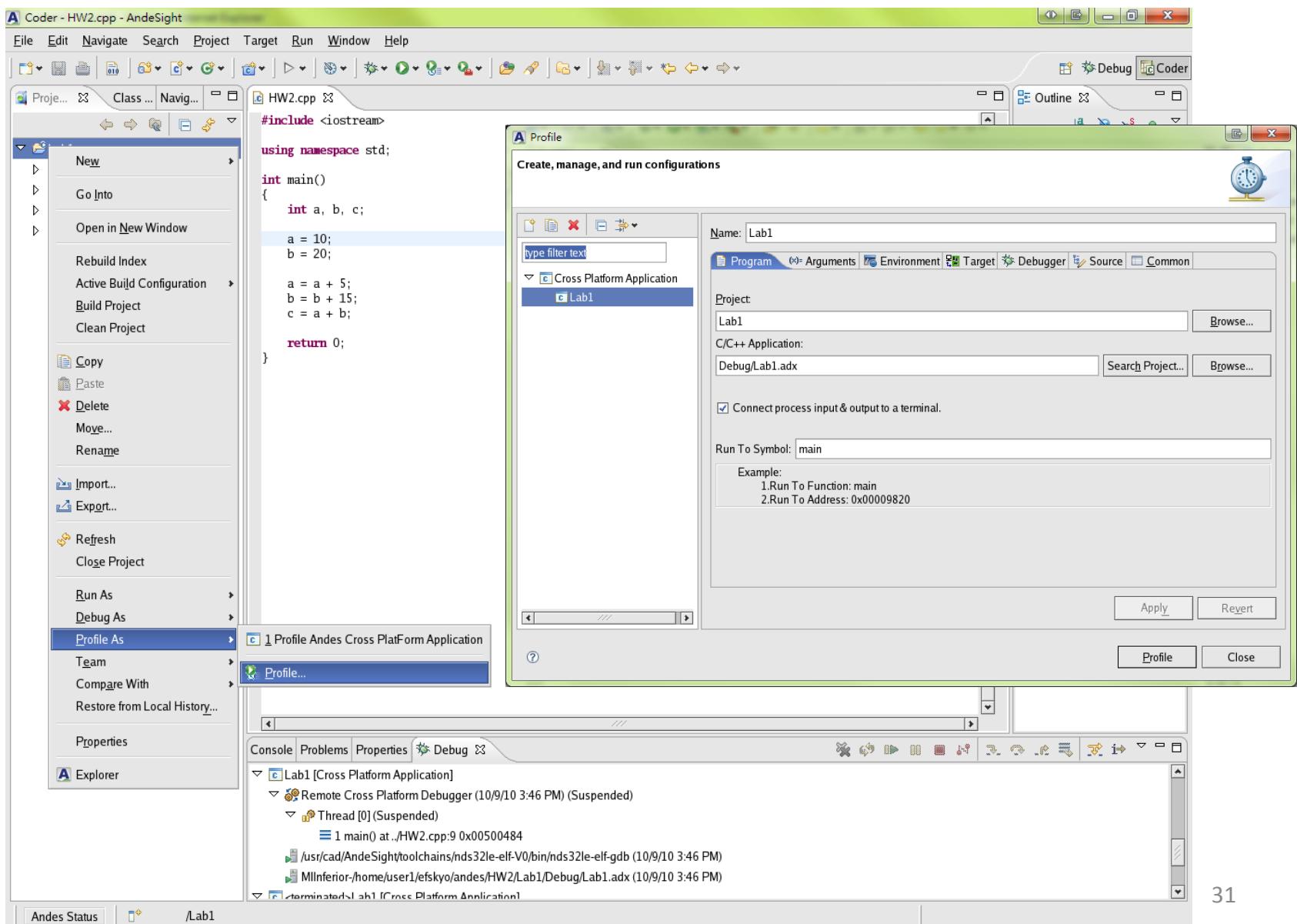
# Switch to Coder



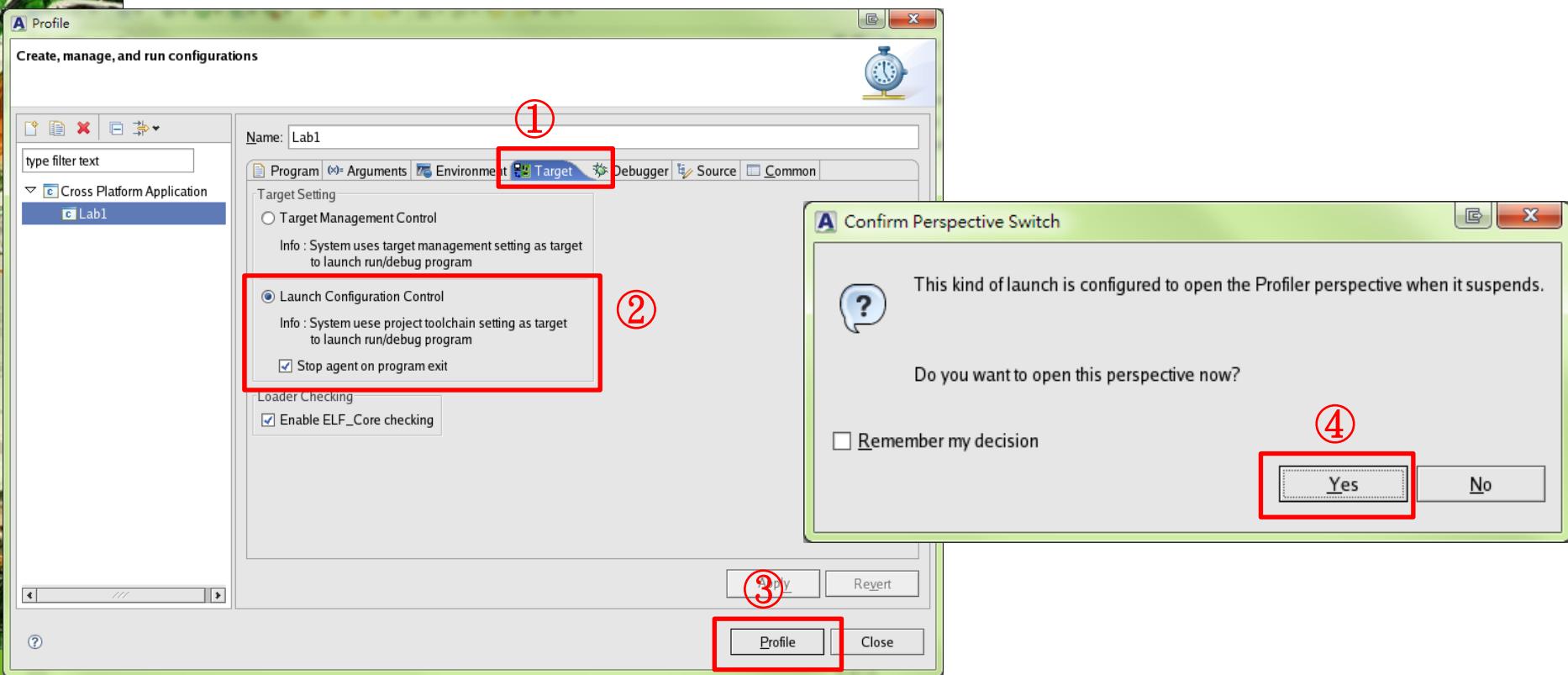
# Profiler



# Profiler(1/6)



# Profiler(2/6)



# Profiler(3/6)

The screenshot shows the AndeSight IDE interface with several windows open:

- Debug View:** Shows a list of processes and threads. A red box highlights the toolbar above the list, which includes icons for Stop, Run, Break, and Step.
- Project Explorer:** Shows the project structure under "Lab1".
- Code Editor:** Displays the C++ code for "HW2.cpp". The line `a = 10;` is highlighted in green.
- Profiling Statistics:** An empty table titled "Total Cycles" with columns: Name, Calls, SelfInsC, Self CycC.
- Target Manager:** Shows a target configuration for "nds32-target-config.conf".
- Console:** Displays the output of the command "OK OK".
- Performance Meter:** A tabbed view showing metrics like Mode, InsC, I\$ Miss, D\$ Miss, BTB Miss, CycC, and Source Cod.

# Profiler(4/6)

The screenshot shows the AndeSight IDE interface with several windows open:

- Profiler - HW2.cpp - AndeSight**: The main window title.
- File Edit Navigate Search Project Advance Target Run Window Help**: The menu bar.
- Debug**: A dropdown menu showing the current session details.
- Project Explore**: Shows the project structure under Lab1.
- HW2.cpp**: The code editor window containing the following C++ code:

```
#include <iostream>
using namespace std;

int main()
{
    int a, b, c;
    a = 10;
    b = 20;

    a = a + 5;
    b = b + 15;
    c = a + b;

    return 0;
}
```
- Performance Meter**: A table showing performance metrics for different threads. The last row is highlighted with a red box.

Mode	InsC	I\$ Miss	D\$ Miss	BTB Miss	CycC	Source Code
2	2	0	0	0	9	N/A
2	2	0	0	0	2	b = 20;
3	3	0	0	0	4	a = a + 5;
3	3	0	0	0	4	b = b + 15;
4	4	0	0	0	5	c = a + b;
1	1	0	0	0	1	return 0;
4	4	0	0	0	4	}
- Profiling Statistics**: A table showing the total cycles (23,510) and a breakdown by function. The first two rows are highlighted with red boxes.

Name	Calls	Self InsC	Self CycC	CPI	Time Percentage
_start	1	9,585	9,585	1.00	40.75%
std::locale::Impl::M_	14	3,318	3,318	1.00	14.11%
memset	2	2,084	2,084	1.00	8.86%
std::locale::id::M_id(	42	742	742	1.00	3.15%
_Unwind_Sjlj_Unreg	42	714	714	1.00	3.04%
std::locale::Impl::lr	1	644	644	1.00	2.74%
__gnu_cxx::__atomic	30	630	630	1.00	2.68%
std::basic_ios<char>	4	552	552	1.00	2.35%
std::numpunct<char>	1	541	541	1.00	2.30%
std::ios_base::ios_ba	4	524	524	1.00	2.23%
_Unwind_Sjlj_Regis	42	462	462	1.00	1.96%
std::locale::locale()	7	364	364	1.00	1.55%
setjmp	42	336	336	1.00	1.43%
__gnu_cxx::__exchange	23	276	276	1.00	1.17%
std::ios_base::Init::Init	1	249	249	1.00	1.06%
std::ios::openmode	4	220	220	1.00	0.94%
- Target Manager**: Shows the target configuration.
- Console**: Displays the output of the current target session.
- Properties**: Shows the properties of the selected target.
- Andes Status**: Shows the status of the Andes processor.

# Profiler(5/6)

The screenshot shows the AndeSight IDE interface with several windows open:

- Profiler - HW2.cpp - AndeSight**: The main window title.
- Debug**: Shows a list of threads, with one thread suspended.
- Project Explore**: Shows the project structure for Lab1.
- Target Manager**: Shows a target configuration.
- Console**: Displays the output of the target manager.
- Performance Meter**: A table showing performance metrics like InsC, I\$ Miss, D\$ Miss, BTB Miss, CycC, and Source Code.
- Variables**: Shows memory variable values.
- Breakpoints**: Shows current breakpoints.
- Profiling Dialog**: A modal dialog box with the following settings:
  - Profiling Level**: Function Level (selected), Branch Summary Information (checked), Cache Summary Information (checked).
  - Branch Level**: Cache Summary Information (unchecked).
  - Pre-process data of profiling views**: Flat View (selected), Flat View Call View, Flat View Call View Timeline/Chart View.
  - Profiling Mode**: Fast Mode (selected).

Red numbered circles indicate specific steps:

- ① Points to the Profiler icon in the toolbar.
- ② Points to the "Branch Summary Information" and "Cache Summary Information" checkboxes in the Profiling Dialog.
- ③ Points to the "OK" button in the Profiling Dialog.

Below the Profiling Dialog, the console output shows:

```
Lab1 [Cross Platform Application] /usr/cad/AndeSight/toolchains/nds32le-elf-V0/bin/nds32le-elf-gdb (10/9/10 4:57 PM)
OK
OK
OK
OK
OK
OK
OK
OK
```

# Profiler(6/6)

The screenshot shows the AndeSight Profiler interface with several windows open:

- Profiler - Lab1 - AndeSight**: The main window showing the Performance Meter, Variables, and Breakpoints tabs. The Performance Meter tab displays memory access statistics (InsC, I\$ Miss, D\$ Miss, BTB Miss, CycC) and source code lines. One line of assembly is highlighted: `b = 20;`
- Debug**: A toolbar with various debug icons.
- Project Explore**: Shows the project structure for Lab1, including Binaries, Includes, Debug, and Profile. The Profile folder contains a file named `Lab1_L1+2+3_10_10_09_17_10_07.pt`, which is highlighted with a red box.
- Performance Meter**: A table showing performance metrics for each function. The columns include Name, Calls, Self InsC, Self CycC, Self InsC/Calls, Self CycC/Calls, Total InsC/Calls, Total CycC/Calls, CPI, and Time Percentage. The row for `__init` has the highest time percentage at 15.29%.
- Target Manager**: Shows target configuration and a console output window. The console output includes GDBAgent startup logs and a message from AndeSight.
- Console**: Displays the same GDBAgent startup logs as the Target Manager.
- Properties**: A tab in the Target Manager window.



# How to extract the useful machine code for our work?

# Run(1/3)

The screenshot shows the AndeSight IDE interface. The main window displays a C++ project structure under the 'HWII' folder, containing files like Binaries, Includes, Debug, Testbench1.cpp, and Testbench2.cpp. The 'Testbench2.cpp' file is open in the central editor area, showing the following code:

```
#include <iostream>
using namespace std;

int main()
{
    int A, B, C, D, E;

    A = 170;
    B = 240;
    C = 15;
    D = 85;

    E = A & B;
    E = E | C;
    E = E ^ D;

    cout << "E = " << E << endl;
    return 0;
}
```

The 'Console' window at the bottom of the interface is highlighted with a red border. It displays the output of the GDBAgent startup process:

```
<terminated> GDBAgent [Local VEP Agent]/usr/cad/AndeSight/GDBAgent/GDBAgent
(GDBAgent) Use license server setting: 10098@vlsicad4
GDBAgent version 1.3.3 - 2009-06-09
(GDBAgent) Succeed in VEP startup.
(GDBAgent) Receive Logout from AndeSight.
```

The 'Andes Status' tab is visible at the bottom left of the IDE.

# Run(2/3)

The screenshot shows the AndeSight Coder IDE interface. The main window displays a C++ file named `Testbench2.cpp` with the following code:

```
#include <iostream>
using namespace std;

int main()
{
    int A, B, C, D, E;

    A = 170;
    B = 240;
    C = 15;
    D = 85;

    E = A & B;
    E = E | C;
    E = E ^ D;

    cout << "E = " << E << endl;
    return 0;
}
```

The IDE includes a Project Explorer on the left showing files like `Testbench1.cpp` and `Testbench2.cpp`. On the right, there's an Outline view showing symbols from the code. At the bottom, a terminal window titled "Console" shows build logs and a red box highlights the "Run" button in the toolbar.

Console output:

```
<terminated> GDBAgent
(GDBAgent) Use lice
GDBAgent version 1.
(GDBAgent) Succeed
(GDBAgent) Receive
1 C-Build [HWII]
✓ 2 <terminated> GDBAgent [Local VEP Agent] /usr/cad/AndeSight/GDBAgent/GDBAgent
3 <terminated> HWII [Cross Platform Application] /usr/cad/AndeSight/toolchains/nds32le-elf-V1/bin/nds32le-elf-gdb (10/12/10 12:11 PM)
4 <terminated> HWII [Cross Platform Application] MIIinferior-/home/user1/efskyo/andes/VSD/HWII/Debug/HWII.adx (10/12/10 12:11 PM)
```

Andes Status:

# Run(3/3)

The screenshot shows a C++ development environment with multiple windows. The main window displays a file named `Testbench1.cpp` containing the following code:

```
#include <iostream>
using namespace std;

int main()
{
    int A, B, C, D, E;

    A = 170;
    B = 240;
    C = 15;
    D = 85;

    E = A & B;
    E = E | C;
    E = E ^ D;

    cout << "E = " << E << endl;
    return 0;
}
```

A red box highlights the assignment of `E` at the bottom of the code. A green arrow points from this highlighted area to the output window. The output window shows the following text:

```
<terminated> HWII [Cross Platform Application] E = 250
```

Another red box highlights the value `E = 250`. A green oval surrounds the `cout` statement in the original code, indicating it is the source of the output. The bottom window shows the same output again.

# Mark the C++ code that we can't use



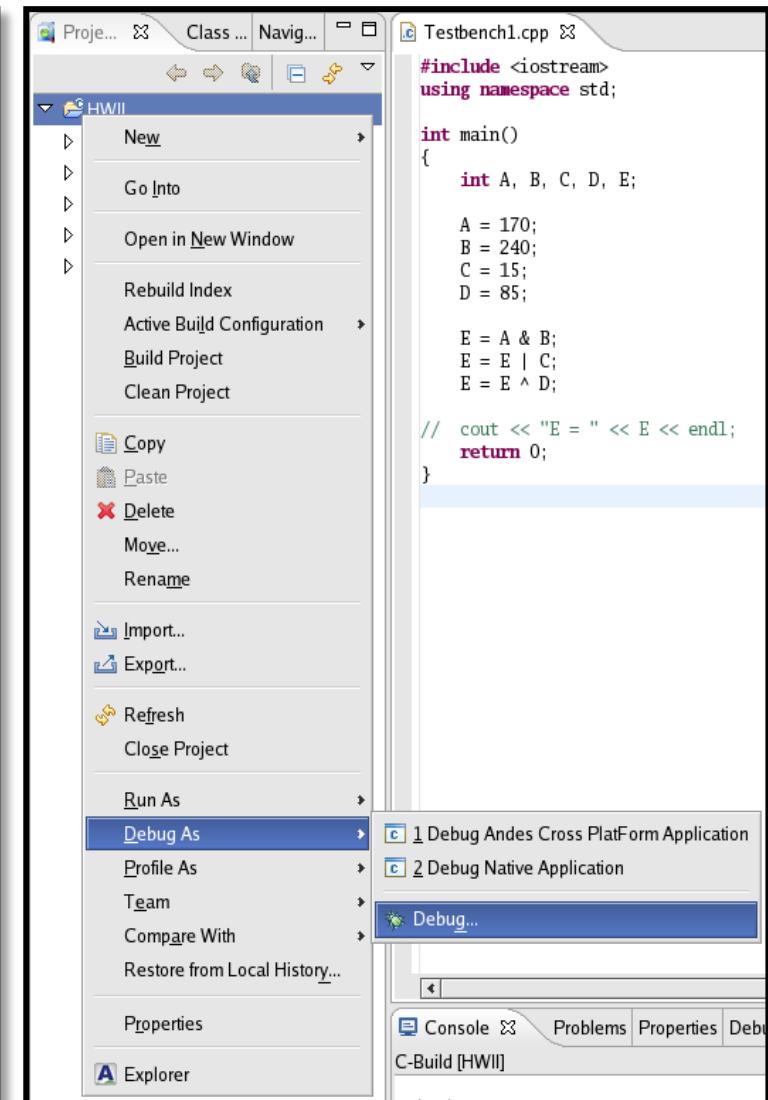
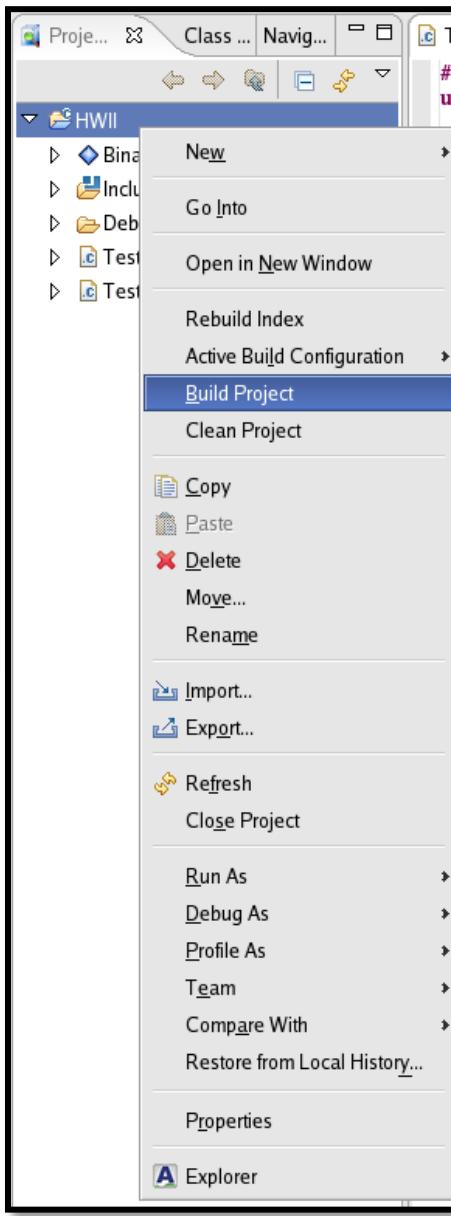
```
#include <iostream>
using namespace std;

int main()
{
    int A, B, C, D, E;

    A = 170;
    B = 240;
    C = 15;
    D = 85;

    E = A & B;
    E = E | C;
    E = E ^ D;

    // cout << "E = " << E << endl;
    return 0;
}
```



```
#include <iostream>
using namespace std;

int main()
{
    int A, B, C, D, E;

    A = 170;
    B = 240;
    C = 15;
    D = 85;

    E = A & B;
    E = E | C;
    E = E ^ D;

    // cout << "E = " << E << endl;
    return 0;
}
```

The code editor shows the same C++ code as the first window. Below the code editor is a toolbar with icons for file operations. To the right of the code editor is a status bar displaying "C-Build [HWII]".



Power

# Assembly code(1/2)

The screenshot shows the AndeSight IDE interface with several windows open:

- Debug - Testbench1.cpp - AndeSight**: The main window showing the project structure and a suspended thread. A red box highlights the "Step Into" button in the toolbar.
- Testbench1.cpp**: The source code window containing the following C++ code:

```
#include <iostream>
using namespace std;

int main()
{
    int A, B, C, D, E;

    A = 170;
    B = 240;
    C = 15;
    D = 85;

    E = A & B;
    E = E | C;
    E = E ^ D;

    // cout << "E = " << E << endl;
    return 0;
}
```

A red box highlights the assembly code area in the Disassembly tab.
- Outline**: Shows the structure of the assembly code, including labels and instructions.
- Disassembly**: Shows the assembly code for the main function. A red box highlights the assembly code area.
- Target Manager**: Shows the target configuration.
- Console**: Shows the console output for the application.
- Andes Status**: Shows the status bar at the bottom.

# Assembly code(2/2)

```
c Testbench1.cpp x
#include <iostream>
using namespace std;

int main()
{
    int A, B, C, D, E;

    A = 170;
    B = 240;
    C = 15;
    D = 85;

    E = A & B;
    E = E | C;
    E = E ^ D;

// cout << "E = " << E << endl;
    return 0;
}
```

Memory Address

0x0050045c
0x00500460
...
...
...
...
...
...
...
...
0x005004b4
0x005004b8

Register	32-bit	Comments
r28	fp	Frame Point
r29	gp	Global Pointer
r30	lp	link pointer
r31	sp	Stack Pointer

{ ★

0x0050045c <main>: addi \$sp,\$sp,#-32  
0x00500460 <main+4>: swi \$fp,[sp+\$4]  
0x00500464 <main+8>: addi \$fp,\$sp,#0  
 int A, B, C, D, E;

A = 170;  
0x00500468 <main+12>: movi \$r0,#170  
0x0050046c <main+16>: swi \$r0,[fp+\$8]  
 B = 240;  
0x00500470 <main+20>: movi \$r0,#240  
0x00500474 <main+24>: swi \$r0,[fp+\$12]  
 C = 15;  
0x00500478 <main+28>: movi \$r0,#15  
0x0050047c <main+32>: swi \$r0,[fp+\$16]  
 D = 85;  
0x00500480 <main+36>: movi \$r0,#85  
0x00500484 <main+40>: swi \$r0,[fp+\$20]

E = A & B;  
0x00500488 <main+44>: lwi \$r1,[fp+\$8]  
0x0050048c <main+48>: lwi \$r0,[fp+\$12]  
0x00500490 <main+52>: and \$r0,\$r1,\$r0  
0x00500494 <main+56>: swi \$r0,[fp+\$24]  
 E = E | C;  
0x00500498 <main+60>: lwi \$r1,[fp+\$24]  
0x0050049c <main+64>: lwi \$r0,[fp+\$16]  
0x005004a0 <main+68>: or \$r0,\$r1,\$r0  
0x005004a4 <main+72>: swi \$r0,[fp+\$24]  
 E = E ^ D;  
0x005004a8 <main+76>: lwi \$r1,[fp+\$24]  
0x005004ac <main+80>: lwi \$r0,[fp+\$20]  
0x005004b0 <main+84>: xor \$r0,\$r1,\$r0  
0x005004b4 <main+88>: swi \$r0,[fp+\$24]

// cout << "E = " << E << endl;  
 return 0;  
} ★

0x005004b8 <main+92>: movi \$r0,#0  
0x005004bc <main+96>: addi \$sp,\$fp,#0  
0x005004c0 <main+100>: lwi \$fp,[fp+\$4]  
0x005004c4 <main+104>: addi \$sp,\$sp,#32  
0x005004c8 <main+108>: ret \$lp

# Extract machine code(1/5)

The screenshot shows the AndeSight debugger interface during a debug session. The top left pane displays the project tree and current thread information. The bottom left pane shows the source code for `Testbench1.cpp`. The right side of the interface contains several windows: a memory dump window showing hex values from address `0x005004b8`, an outline window, a disassembly window showing assembly instructions, and a registers window. A large orange starburst annotation points to the memory dump window with the text "End : 0x005004b8".

Memory Dump (Hex View) at Address `0x005004b8`:

Address	0 - 3	4 - 7	8 - B	C - F
005004B0	40008003	140E0006	44000000	31FE0000
005004C0	05CE0001	51FF8020	44007820	31FFFFE0
005004D0	15CF8007	51CF8018	15EE0000	140E0002
005004E0	141E0003	041E0003	4400FFFF	4C10401A
005004F0	041E0002	44000001	4C104014	46000548
00500500	580003AC	4900011C	46000500	58000534
00500510	44100000	46200548	5821031C	4901B640
00500520	0DEE7FFF	51FE0000	05CE0002	51FF800C

Disassembly:

```
0x00500498 <main+60>: lwi $r1,[fp+#16]
0x0050049c <main+64>: lwi $r0,[fp+#16]
0x005004a0 <main+68>: or $r0,$r1,$r0
0x005004a4 <main+72>: swi $r0,[fp+#24]
    E = E ^ D;
0x005004a8 <main+76>: lwi $r1,[fp+#24]
0x005004ac <main+80>: lwi $r0,[fp+#20]
0x005004b0 <main+84>: xor $r0,$r1,$r0
0x005004b4 <main+88>: swi $r0,[fp+#24]

// cout << "E = " << E << endl;
    return 0;
0x005004b8 <main+92>: movi $r0,#0
}
0x005004bc <main+96>: addi $sp,$fp,#0
0x005004c0 <main+100>: lwi $fp,[fp+#4]
0x005004c4 <main+104>: addi $sp,$sp,#32
0x005004c8 <main+108>: ret $lp
```

Target Manager: VEP:[vlsicad4]nds32-target-config.conf

Console: HWII [Cross Platform Application] Mlinferior-/home/user1/efskyo/andes/VSD/HWII/Debug/HWII.adx (10/12/10 4:42 PM)

Andes Status

# Extract machine code(2/5)

The screenshot shows the AndeSight IDE interface during a debugging session. The top menu bar includes File, Edit, Navigate, Search, Project, Target, Run, Window, Help, and tabs for Debug, Coder, Memory, Registers, Modules, and Renderings. A red box labeled ② highlights the Memory tab. A red box labeled ③ highlights the Monitor button in the left sidebar. A red box labeled ⑥ highlights the close button in the top right corner of the main window.

The left pane displays the project structure for 'Testbench1.cpp' and its contents:

```
#include <iostream>
using namespace std;

int main()
{
    int A, B, C, D, E;
    A = 170;
    B = 240;
    C = 15;
    D = 85;

    E = A & B;
    E = E | C;
    E = E ^ D;

    // cout << "E = " << E << endl;
    return 0;
}
```

A yellow box labeled ④ highlights the 'Paste' button in the code editor. A red box labeled ⑤ highlights the 'OK' button in the 'Monitor Memory' dialog box. The dialog box contains the address '0x0050045d'.

The right pane shows the memory dump for address 0x0050045c. An orange starburst labeled 'Start : 0x0050045c' points to the first memory location. A red box labeled ① highlights the address '0x0050045c'. A red box labeled ② highlights the 'Copy' button. The memory dump table has columns for Address, 0 - 3, 4 - 7, 8 - B, and C - F.

Address	0 - 3	4 - 7	8 - B	C - F
00500450	05CE0002	51FF802C	4A007820	51FFFFE0
00500460	15CF8001	51CF8000	440000AA	140E0002
00500470	440000F0	140E0003	4400000F	140E0004
00500480	44000055	140E0005	041E0002	040E0003
00500490	40008002	140E0006	041E0006	040E0004
005004A0	40008004	140E0006	041E0006	040E0005
005004B0	40008003	140E0006	44000000	51FE0000
005004C0	05CE0001	51FF8020	4A007820	51FFFFE0

The bottom pane shows the Target Manager and Console. The Target Manager lists 'VEP:[vlsicad4]nds32-target-config.conf'. The Console output is: 'HWII [Cross Platform Application]/usr/cad/AndeSight/toolchains/nds32le-elf-V1/bin/nds32le-elf-gdb (10/13/10 10:55 AM) No symbol "endl" in current context.'

**Logo:** VLSI Design LAB

# Extract machine code(3/5)

A Debug - Testbench1.cpp - AndeSight

File Edit Navigate Search Run Project Target Window Help

Variables Breakpoints Memory Monitors

0x00500450 0x00500454 0x00500458 0x0050045C

0x0050045c : 0x50045C <Hex>

Address	0 - 3	4 - 7	8 - B	C - F
00500450	05CE0002	51FF802C	4A007820	51FFFFFF
00500460	15CF8001	51CF8000	440000AA	140E0002
00500470	440000F0	140E0003	4400000F	140E0004
00500480	44000055	140E0005	041E0002	040E0003
00500490	40008002	140E0006	041E0006	040E0004
005004A0	40008004	140E0006	041E0006	040E0005
005004B0	40008003	140E0006	44000000	51FFFFFF
005004C0	05CE0001	51FF8020	4A007820	51FFFFFF
005004D0	15CF8007	51CF8018	15EE0000	140E0002
005004E0	141E0003	041E0003	4400FFFF	4C10401A
005004F0	041E0002	44000001	4C104014	46000548
00500500	580003AC	4900011C	46000500	58000534
00500510	44100000	46200548	5821031C	4901B640
00500520	0DEE7FFF	51FE0000	05CE0002	51FF800C
00500530	4A007820	51FFFFFF	15CF8007	51CF8018
00500540	15EE0000	140E0002	46000548	580003AC
00500550	490000A2	0DEE7FFF	51FE0000	05CE0002
00500560	51FF800C	4A007820	51FFFFFF	15CF8007
00500570	51CF8018	15EE0000	44000001	4410FFFF
00500580	49FFFA6	0DEE7FFF	51FE0000	05CE0002

Start : 0x0050045c

End : 0x005004b8

Andes Status

# Extract machine code(4/5)

The screenshot shows the AndesSight debugger interface with the title bar "A Debug - Testbench1.cpp - AndesSight". The menu bar includes File, Edit, Navigate, Search, Run, Project, Target, Window, Help, and several tool icons. The toolbar has buttons for file operations like Open, Save, and Print, along with symbols for Breakpoints, Memory, Registers, and Modules.

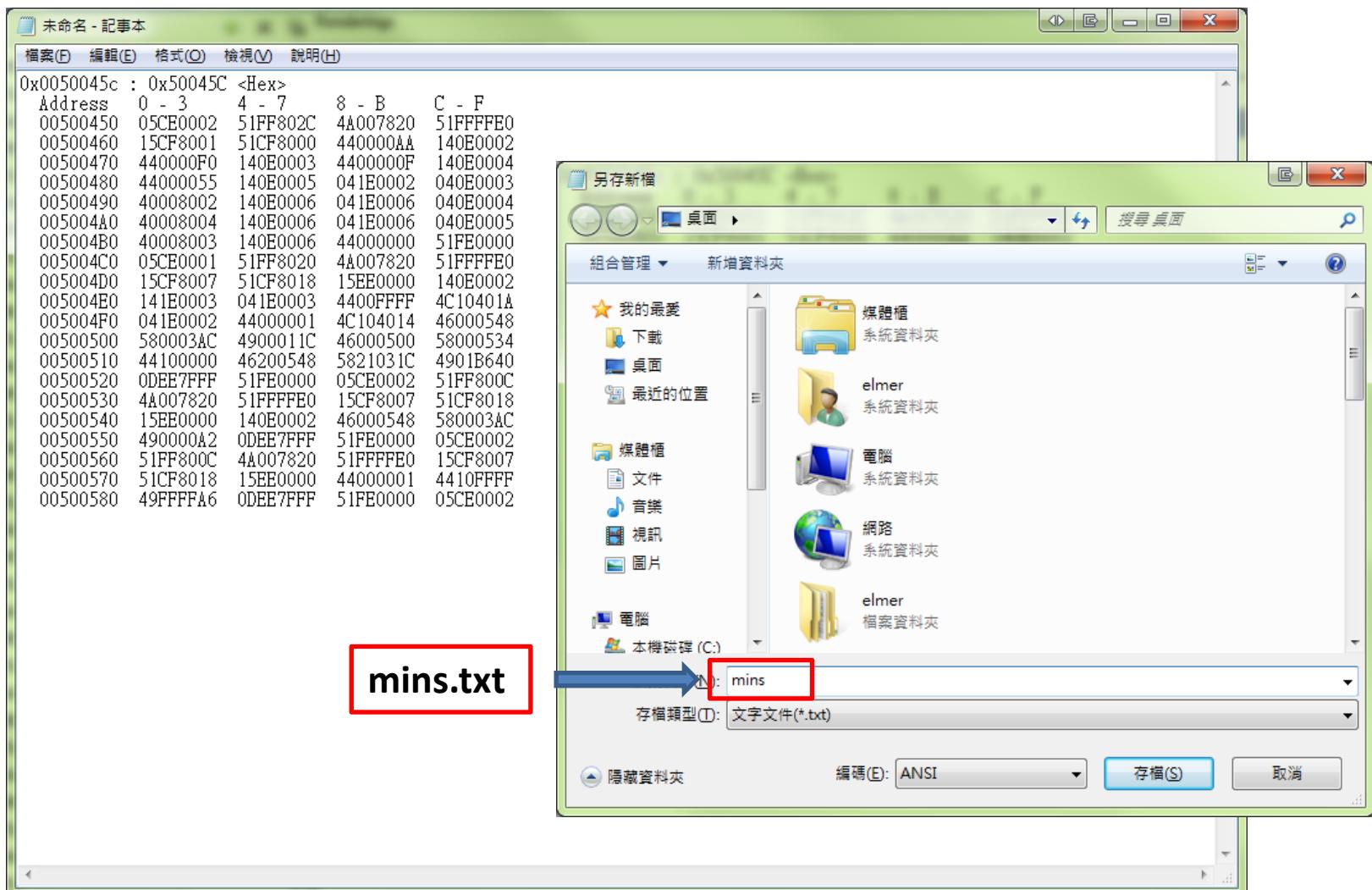
The main window displays a memory dump titled "0x0050045c : 0x50045C <Hex>". The dump is organized into columns: Address, 0 - 3, 4 - 7, 8 - B, C - F, and a blank column. The first few rows of data are:

Address	0 - 3	4 - 7	8 - B	C - F	
00500450	05CE0002	51FF802C	4A007820	51FF	
00500460	15CF8001	51CF8000	440000AA	140E	
00500470	440000F0	140E0003	4400000F	140E	
00500480	44000055	140E0005	041E0002	040E	
00500490	40008002	140E0006	041E0006	040E	
005004A0	40008004	140E0006	041E0006	040E	
005004B0	40008003	140E0006	44000000	51FEC	
005004C0	05CE0001	51FF8020	4A007820	51FF	
005004D0	15CF8007	51CF8018	15EE0000	140E	
005004E0	141E0003	041E0003	4400FFFF	4C104	
005004F0	041E0002	44000001	4C104014	46000	
00500500	580003AC	4900011C	46000500	58000	
00500510	44100000	46200548	5821031C	4901E	
00500520	0DEE7FFF	51FE0000	05CE0002	51FF	
00500530	4A007820	51FFFFFF	15CF8007	51CF	
00500540	15EE0000	140E0002	46000548	58000	
00500550	490000A2	0DEE7FFF	51FE0000	05CE	
00500560	51FF800C	4A007820	51FFFFFF	15CF	
00500570	51CF8018	15EE0000	44000001	4410F	
00500580	49FFFA6	0DEE7FFF	51FE0000	05CE0002	

A context menu is open over the first row of the dump, listing options: Add Rendering, Remove Rendering, Reset to Base Address..., Go to Address..., Format..., Previous Page, Next Page, Resize to Fit, Hide Address Column, Copy To Clipboard (which is highlighted), Print, Properties, and Input Methods.

The left sidebar lists memory locations from 0x00500450 to 0x0050045c. The bottom status bar shows "Andes Status".

# Extract machine code(5/5)



# Format Transform(1/2)

## □ transform.exe

```
Please select b<binary> or h<hexadecimal> : b
Monitor 0x0050045c : 0x50045C <Hex>
Address 0 - 3    4 - 7    8 - B    C - F
500450 5ce0002 51ff802c 4a007820 51ffffe0
500460 15cf8001 51cf8000 440000aa 140e0002
500470 440000f0 140e0003 4400000f 140e0004
500480 44000055 140e0005 41e0002 40e0003
500490 40008002 140e0006 41e0006 40e0004
5004a0 40008004 140e0006 41e0006 40e0005
5004b0 40008003 140e0006 44000000 51fe0000
5004c0 5ce0001 51ff8020 4a007820 51ffffe0
5004d0 15cf8007 51cf8018 15ee0000 140e0002
5004e0 141e0003 41e0003 4400ffff 4c10401a
5004f0 41e0002 44000001 4c104014 46000548
500500 580003ac 4900011c 46000500 58000534
500510 44100000 46200548 5821031c 4901b640
500520 dee7fff 51fe0000 5ce0002 51ff800c
500530 4a007820 51ffffe0 15cf8007 51cf8018
500540 15ee0000 140e0002 46000548 580003ac
500550 490000a2 dee7fff 51fe0000 5ce0002
500560 51ff800c 4a007820 51ffffe0 15cf8007
500570 51cf8018 15ee0000 44000001 4410ffff
500580 49ffffa6 dee7fff 51fe0000 5ce0002
請按任意鍵繼續 . . . -
```

mins.txt

transform

mins.prog

# Format Transform(2/2)

mins.prog

未命名 - 記事本

檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)

```
0x0050045c : 0x50045C <Hex>
Address 0 - 3 4 - 7 8 - B C - F
00500450 05CE0002 51FF802C 4A007820 51FFFFE0
00500460 15CF8001 51CF8000 440000AA 140E0002
00500470 440000F0 140E0003 4400000F 140E0004
00500480 44000055 140E0005 041E0002 040E0003
00500490 40008002 140E0006 041E0006 040E0004
005004A0 40008004 140E0006 041E0006 040E0005
005004B0 40008003 140E0006 44000000 51FE0000
005004C0 05CB0001 51FF8020 4A007820 51FFFFE0
005004D0 15CF8007 51CF8018 15EE0000 140E0002
005004E0 141B0003 041E0003 4400FFFF 4C10401A
005004F0 041B0002 44000001 4C104014 46000548
00500500 580003AC 4900011C 46000500 58000534
00500510 44100000 46200548 5821031C 4901B640
00500520 ODEE7FFF 51FE0000 05CE0002 51FF800C
00500530 4A007820 51FFFFE0 15CF8007 51CF8018
00500540 15EE0000 140E0002 46000548 580003AC
00500550 490000A2 ODEE7FFF 51FE0000 05CE0002
00500560 51FF800C 4A007820 51FFFFE0 15CF8007
00500570 51CF8018 15EE0000 44000001 4410FFFF
00500580 49FFFA6 ODEE7FFF 51FE0000 05CE0002
```

mins.txt

1 @500450  
2 0000\_0101\_1100\_1110\_0000\_0000\_0000\_0010  
3 @500454  
4 0101\_0001\_1111\_1111\_1000\_0000\_0010\_1100  
5 @500458  
6 0100\_1010\_0000\_0000\_0111\_1000\_0010\_0000  
7 @50045c  
8 0101\_0001\_1111\_1111\_1111\_1111\_1110\_0000  
9 @500460  
10 0001\_0101\_1100\_1111\_1000\_0000\_0000\_0001  
11 @500464  
12 0101\_0001\_1100\_1111\_1000\_0000\_0000\_0000  
13 @500468  
14 0100\_0100\_0000\_0000\_0000\_0000\_1010\_1010  
15 @50046c  
16 0001\_0100\_0000\_1110\_0000\_0000\_0000\_0010  
17 @500470  
18 0100\_0100\_0000\_0000\_0000\_0000\_1111\_0000  
19 @500474  
20 0001\_0100\_0000\_1110\_0000\_0000\_0000\_0011  
21 @500478  
22 0100\_0100\_0000\_0000\_0000\_0000\_0000\_1111  
23 @50047c  
24 0001\_0100\_0000\_1110\_0000\_0000\_0000\_0100

binary

1 @500450  
2 5ce0002  
3 @500454  
4 51ff802c  
5 @500458  
6 4a007820  
7 @50045c  
8 51ffffe0  
9 @500460  
10 15cf8001  
11 @500464  
12 51cf8000  
13 @500468  
14 440000aa  
15 @50046c  
16 140e0002  
17 @500470  
18 440000f0  
19 @500474  
20 140e0003  
21 @500478  
22 4400000f  
23 @50047c  
24 140e0004

hexadecimal

# Reference

- Andes Technology Corporation, “AndeSight\_v1.3.3\_User\_Manual.pdf”, June. 2009.
- Andes Technology Corporation, “AndeStar\_ISA\_v1.3.pdf”, May 2009.
- Andes Technology Corporation, “Embedded System Introduction.ppt”, December 2009.
- Andes Technology Corporation, “ AndESLive modeling training.ppt”, May 2010.



**Thanks for your participation  
and attendance ! !**