

AndeSight™ v1.4-MCU



Agenda



- ❖ Design Flow
- ❖ AndESLive™
- ❖ Profiling
- ❖ AndeSight™ MCU Features

Agenda

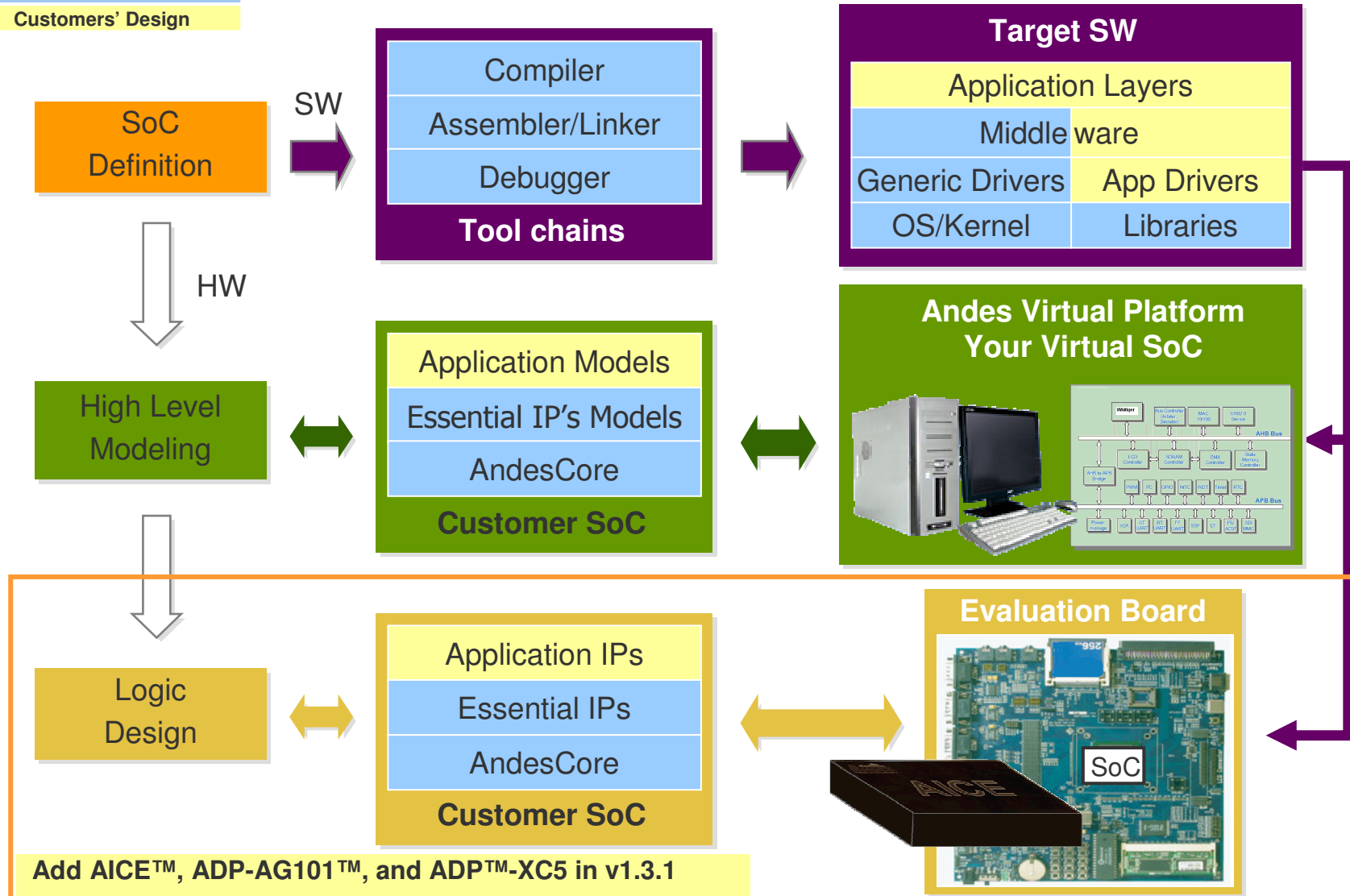


- ❖ Design Flow
- ❖ AndESLive™
- ❖ Profiling
- ❖ AndeSight™ MCU Features

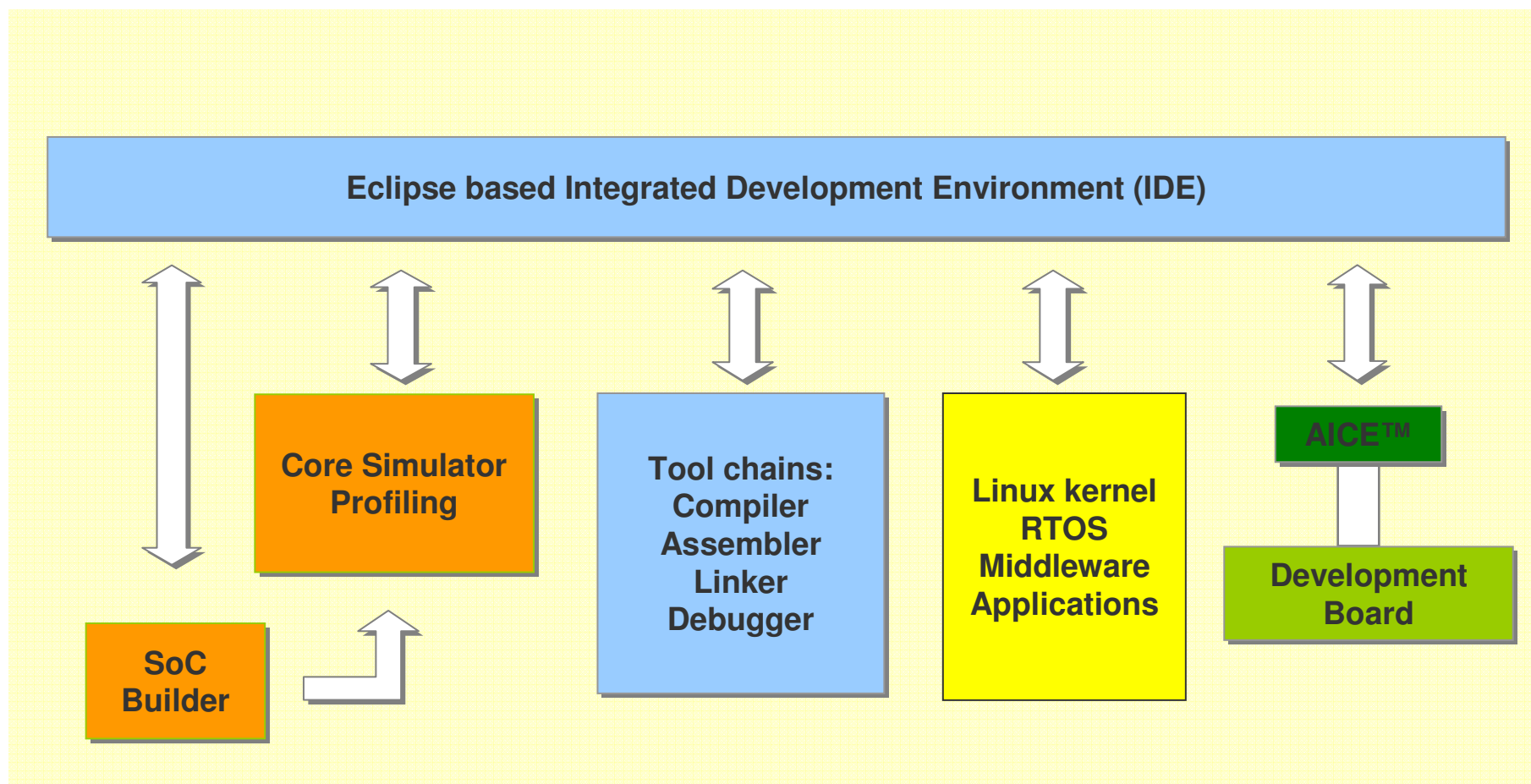
SoC Development Flow



Andes/Partners' solution
Customers' Design



Andes Total SW Solution



Andes SW Solution: AndESLive™ + AndeSight™ + AndeSoft™ + AICE™ + AndeShape™

Agenda



- ❖ Design Flow
- ❖ AndESLive™
- ❖ Profiling
- ❖ AndeSight™ MCU Features

AndESLive™ Highlights



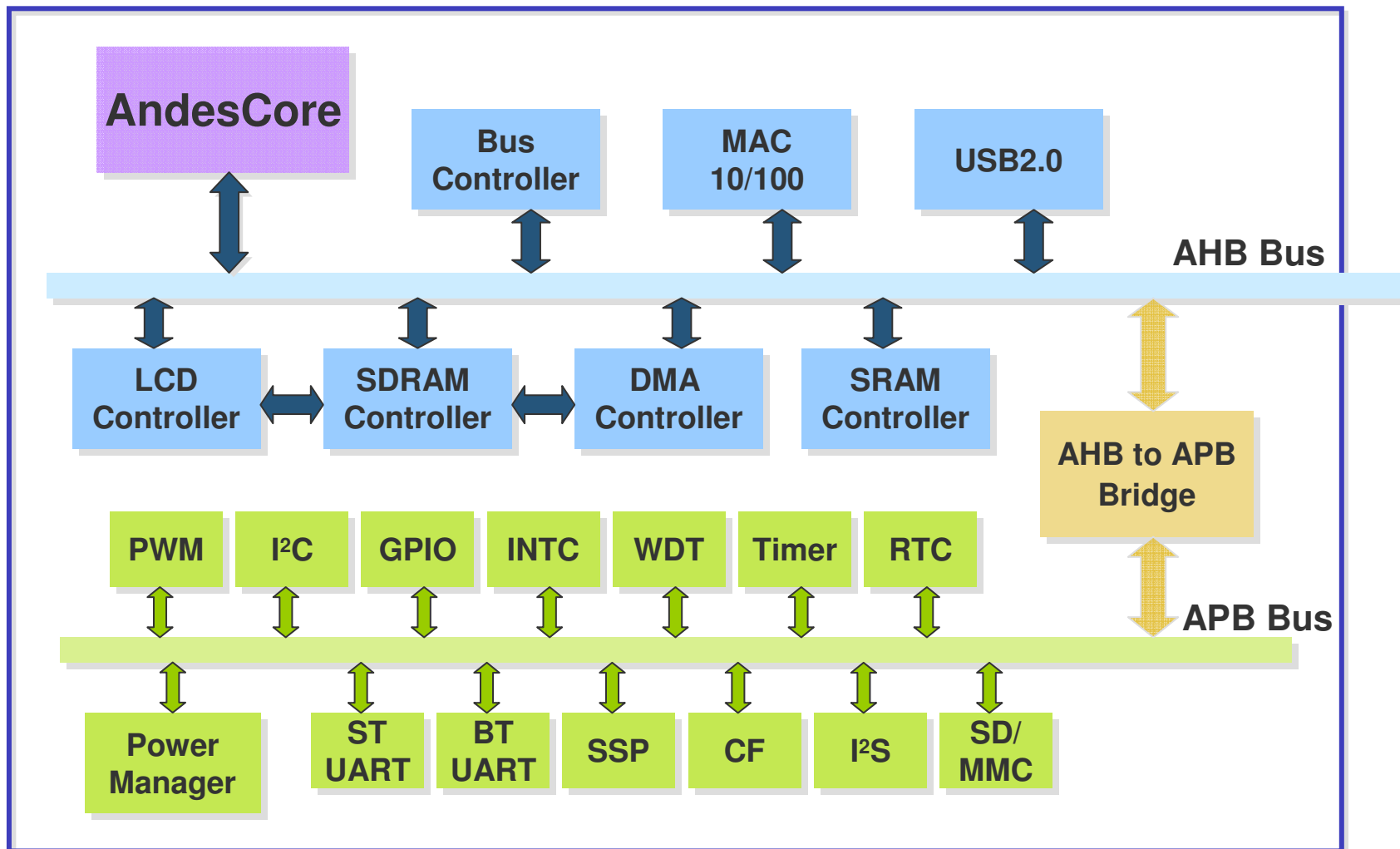
❖ SoC builder

- Pre-defined models of AndesCore™, peripheral IPs, bus
- Drag-n-Drop to add components and/or bus from Palette
- Two clicks to connect pins or ports
- Table driven and setting of components properties, memory map, and IRQ

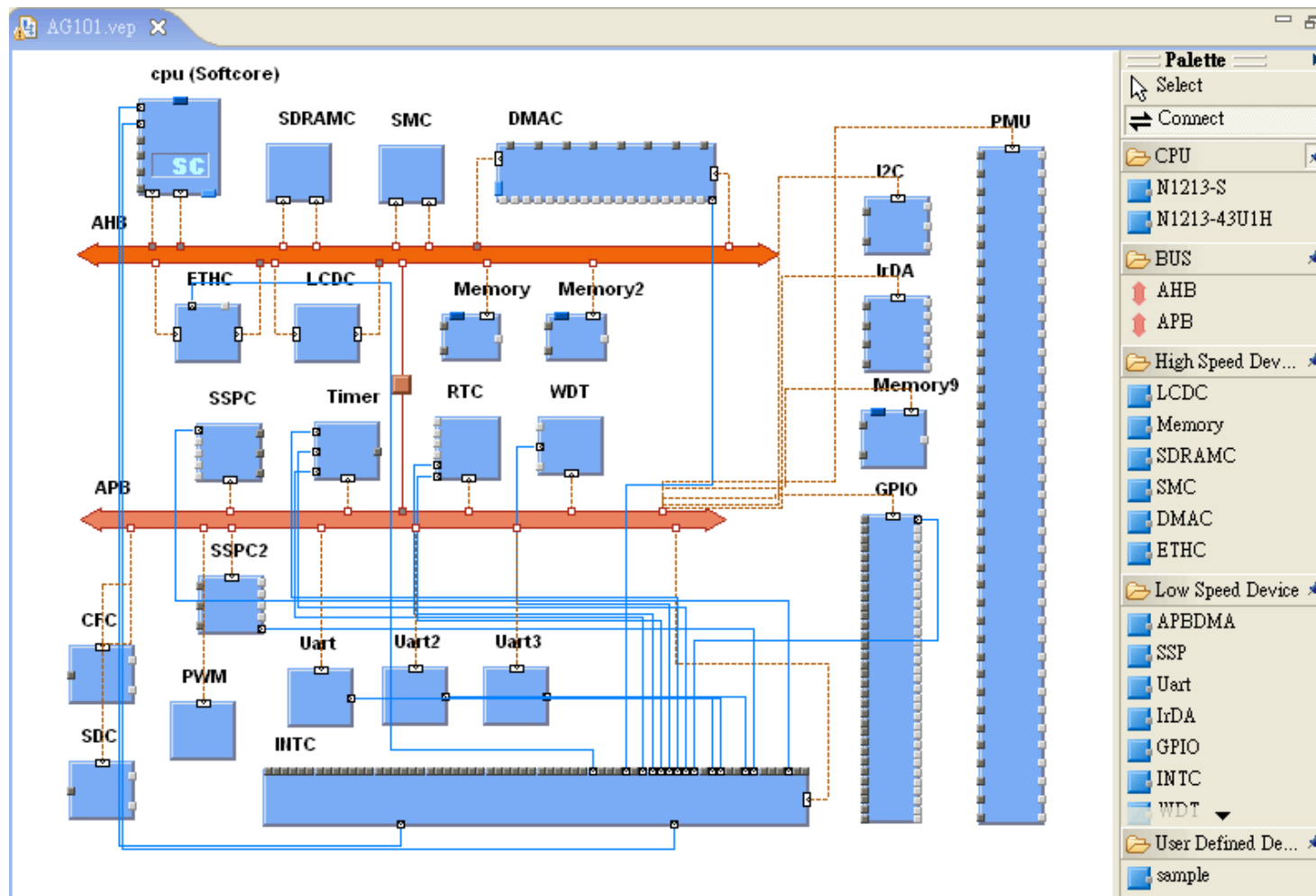
❖ Simulator

- Cycle-based and instruction-based simulator
- Fast to run software application
- Visibility of debugging and profiling data
- Simulation of I/O devices

AndeShape™ Platform SoC: AG101



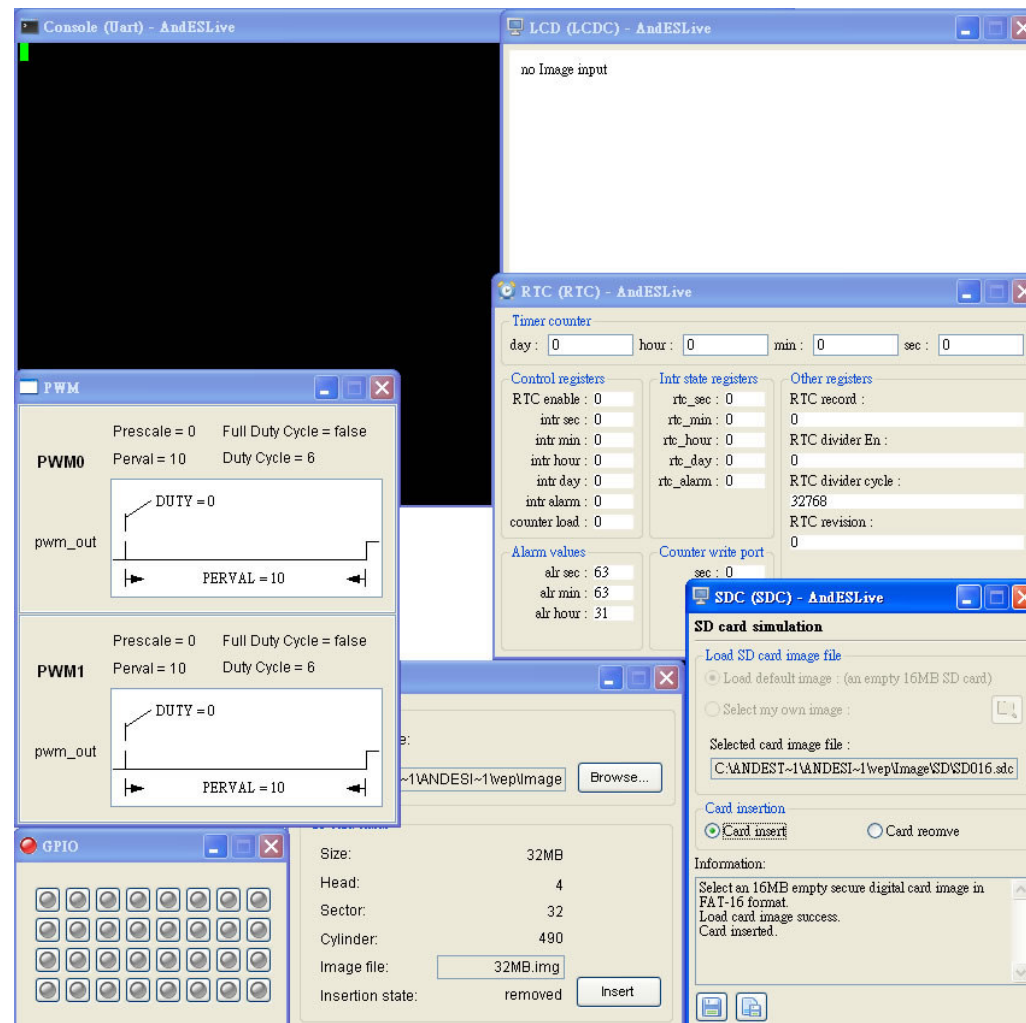
AndESLive™ Models



AndESLive™ Virtual IO



- ❖ UART
- ❖ LCD
- ❖ RTC
- ❖ SDC
- ❖ PWM
- ❖ GPIO
- ❖ CFC



Agenda



- ❖ Design Flow
- ❖ AndESLive™
- ❖ Profiling
- ❖ AndeSight™ MCU Features

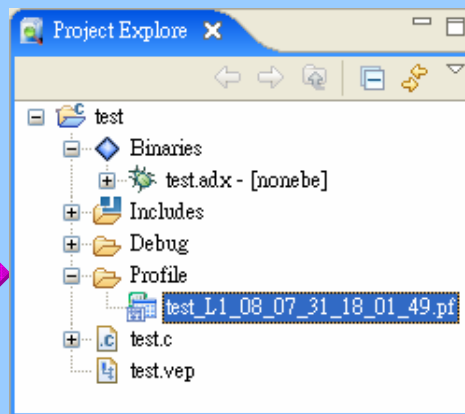
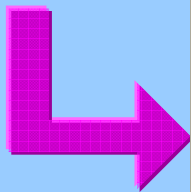
Profiling



AndeSight IDE



Trigger Profiling



Name	Line	Call	Self UseC	Self UseCAll	Self UseCAll	Total Call L	Total Call	CPI	Time Percentage
main	1	37	454	37.00	454.00	91.00	1,192.00	13.54	0.09%
main	1	190	747	190.00	747.00	903.00	5,968.00	393	0.75%
main	1	21	455	21.00	455.00	182.00	2,480.00	21.87	0.07%
main	1	67	620	67.00	620.00	141.00	2,025.00	17.73	0.10%
main	1	15	267	15.00	267.00	39.00	643.00	17.88	0.03%
main	1	25	453	25.00	453.00	25.00	453.00	18.12	0.06%
main	1	31	460	31.00	460.00	34.00	644.00	14.64	0.08%
main	1	55	748	55.00	748.00	1,016.00	7,151.00	13.68	0.75%
main	1	70	1,389	70.00	1,389.00	70.00	1,389.00	16.76	0.07%
main	1	31	231	15.50	115.50	55.00	729.50	7.45	0.54%
main	1	24	279	24.00	279.00	115.00	1,471.00	11.62	0.05%
main	1	27	469	26.50	224.50	30.00	275.50	7.88	0.05%
main	1	24	376	24.00	376.00	24.00	376.00	15.67	0.08%
main	1	22	273	22.00	273.00	64.00	1,084.00	12.41	0.04%
main	1	30	126	15.00	63.00	99.00	474.00	4.20	0.30%
main	1	15	274	15.00	274.00	15.00	274.00	18.27	0.04%
main	1	27	265	27.00	265.00	42.00	478.00	7.59	0.04%
main	1	9	179	9.00	179.00	9.00	179.00	19.89	0.42%
main	1	147	762	147.00	762.00	147.00	762.00	5.18	0.79%
main	1	25	367	25.00	367.00	117.00	1,325.00	14.68	0.08%
main	1	21	196	21.00	196.00	79.00	944.00	9.33	0.46%
main	1	26	369	26.00	369.00	95.00	1,552.00	14.19	0.08%
main	1	10	185	10.00	185.00	2,999.00	21,548.00	18.50	0.41%
main	1	34	476	34.00	476.00	74.00	1,145.00	14.00	0.12%
main	1	156	1,698	156.00	1,698.00	5,009.00	42,995.00	10.88	0.90%
main	1	42	611	42.00	611.00	42.00	611.00	13.31	0.50%
main	1	80	913	80.00	913.00	80.00	913.00	11.41	0.14%
main	1	7	177	7.00	177.00	7.00	177.00	25.29	0.41%
main	1	12	270	12.00	270.00	12.00	270.00	25.50	0.05%
main	1	17	268	17.00	268.00	17.00	268.00	15.76	0.03%
main	1	113	1,453	113.00	1,453.00	2,010.00	14,577.00	12.86	0.40%
main	1	90	1,778	90.00	1,778.00	2,710.00	16,627.00	14.28	0.59%
main	1	7	95	7.00	95.00	7.00	95.00	13.57	0.02%
main	1	16	269	16.00	269.00	50.00	748.00	16.81	0.03%
main	1	23	281	23.00	281.00	69.00	1,149.00	12.22	0.05%
main	1	66	1,114	133.00	985.00	985.00	1,610.00	6.48	1.16%



[name]_[profiling level]_[yy_mm_dd_hh_min_sec].pf

Profiling data preparation



Profiling Analysis Engine

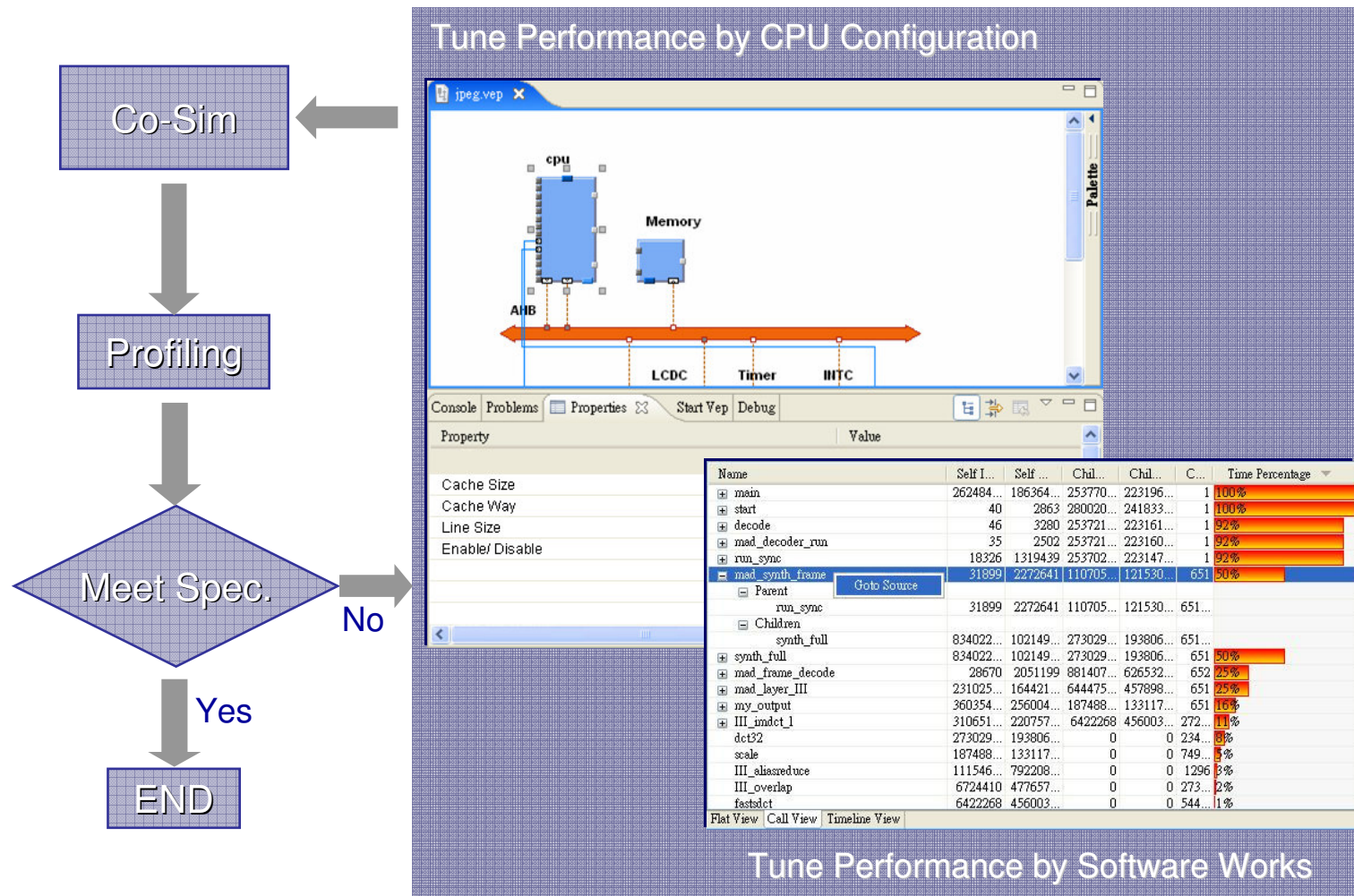
AndESLive Simulator

Profiling Options



- ❖ Function Level
 - Pure function profiling without branch and cache information
 - With Branch Summary
 - With Cache Summary
 - With Branch and Cache Summary
- ❖ Branch Level
 - Pure branch profiling without cache information
 - With Cache Summary
- ❖ Extensive Profiling Views
 - Flat View
 - Call View
 - Timeline View
 - Chart View
- ❖ Real Time Profiler Views
 - Performance Meter View
 - Profiling Statistics View
- ❖ C and C++ Support
- ❖ Fast Mode and Extensive Mode
- ❖ Goto Source

Performance Tuning



Tune Performance by Profiler



Profile Result of 8KB I\$/D\$

Name	Calls	Self InsC	Self CycC	Self InsC...	Self CycC...	Total Call ...	Total Call ...	CPI	Tim
jpeg_idct_islow	1,800	7,349,335	12,672,632	4,082.96	7,040.35	4,082.96	7,040.35	1.72	24.43%
ycc_rgb_convert	240	3,935,760	8,298,381	16,399.00	34,576.59	16,399.00	34,576.59	2.11	16.00%
decode_mcu	300	4,184,926	8,007,311	13,949.75	26,691.04	20,106.21	37,404.26	1.91	15.43%
h2v2_fancy_upsample	240	4,545,600	7,467,432	18,940.00	31,114.30	18,940.00	31,114.30	1.64	14.39%
put_pixel_rows	240	2,316,480	5,380,615	9,652.00	22,419.23	9,733.91	22,981.75	2.32	10.37%
memcpy	13	164,177	2,582,023	12,629.00	198,617.15	12,629.00	198,617.15	15.73	4.98%
jpeg_fill_bit_buffer	9,650	1,269,957	2,036,858	131.60	211.07	182.17	303.19	1.60	3.93%
memset	313	212,993	1,664,279	680.49	5,317.19	680.49	5,317.19	7.81	3.21%
my_fread	12	836,028	1,516,063	69,669.00	126,338.58	69,669.00	126,338.58	1.81	2.92%
decompress_onepass	15	155,768	442,768	10,384.53	29,517.87	916,438.40	1,730,121.67	2.84	0.85%
cache_clean_d	241	374,514	411,877	1,554.00	1,709.03	1,554.00	1,709.03	1.10	0.79%
inex_huff_decode	960	80,034	288,154	91.88	297.37	105.60	320.21	3.24	0.56%

Profile Result of 64KB I\$/D\$

Name	Calls	Self InsC	Self CycC	Self InsC...	Self CycC...	Total Call ...	Total Call ...	CPI	Time Percentage
jpeg_idct_islow	1,800	7,349,335	10,739,885	4,082.96	5,966.60	4,082.96	5,966.60	1.46	23.33%
h2v2_fancy_upsample	240	4,545,600	6,649,947	18,940.00	27,708.11	18,940.00	27,708.11	1.46	14.45%
ycc_rgb_convert	240	3,935,760	6,509,230	16,399.00	27,121.79	16,399.00	27,121.79	1.65	14.14%
decode_mcu	300	4,184,926	6,006,169	13,949.75	20,020.56	20,106.21	29,357.05	1.44	13.05%
put_pixel_rows	240	2,316,480	4,880,829	9,652.00	20,336.79	9,733.91	20,473.06	2.11	10.60%
cache_clean_d	241	2,965,746	2,970,988	12,306.00	12,327.75	12,306.00	12,327.75	1.00	6.45%
memcpy	13	164,177	2,579,164	12,629.00	198,397.23	12,629.00	198,397.23	15.71	5.60%
jpeg_fill_bit_buffer	9,650	1,269,957	1,856,013	131.60	192.33	182.17	276.39	1.46	4.03%
memset	313	212,993	1,490,756	680.49	4,762.80	680.49	4,762.80	7.00	3.24%
my_fread	12	836,028	1,388,828	69,669.00	115,735.67	69,669.00	115,735.67	1.66	3.02%
decompress_onepass	15	155,768	245,457	10,384.53	16,363.80	916,438.40	1,415,277.20	1.58	0.53%

Agenda



- ❖ Design Flow
- ❖ AndESLive™
- ❖ Profiling
- ❖ AndeSight™ MCU Features

AndeSight™ MCU Installation



- ❖ Installation package
 - AndeSight™
 - AndESLive™
 - Toolchains
 - Cygwin
 - Demo cases
- ❖ Supported Core
 - N10 Family - N1033A-S; N1033-S
 - N9 Family - N903A-S; N903-S
- ❖ Optimum System requirements
 - Windows XP Professional (SP2/SP3)
 - 1 GB of RAM for minimum requirement; 2GB is recommended
 - Host disk space (1.4GB for V1.4 MCU)
 - N10 family: 1.04 GB
 - N9 family: 1.4 GB

License Control



❖ License control

- License file is different for MCU and Standard version
 - UI will be different (SoC builder, Target for example) according to the license file
- Node-lock control
 - No Flexnet license server is required
 - Lock on MAC address or HDD serial number
- License management tool enhancement (LicenseMan)
 - License application
 - License deploy

Start-up Code Development



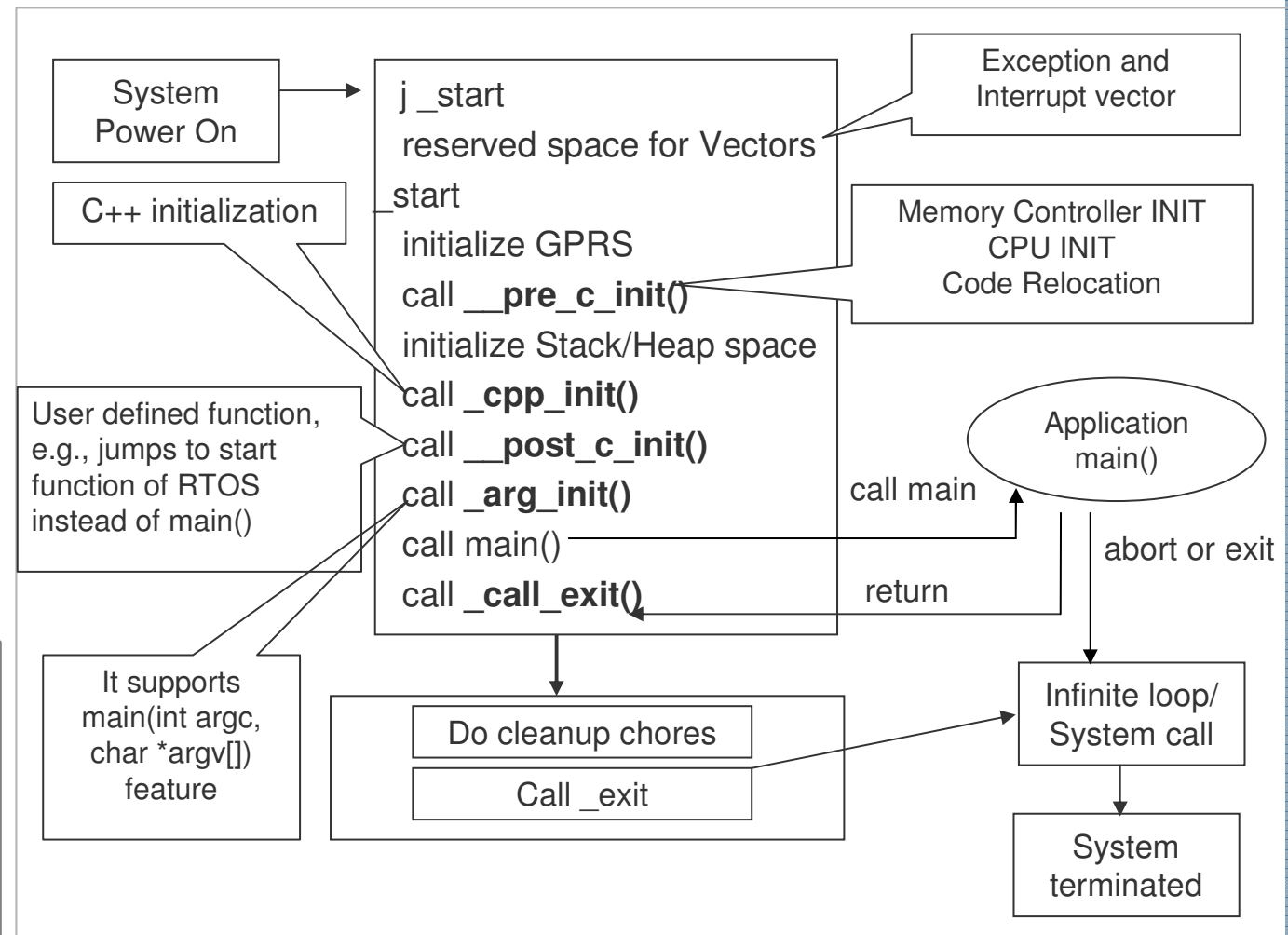
- ❖ The file crt0.o which provided by GNU tool chain is processor-dependent and will be executed by the processor core immediately after system power-on.
- ❖ In Andes version, some weak symbols are defined in order to customize your program easily depending on the features you need.
- ❖ These weak symbols are
 - **__pre_c_init, __post_c_init, _cpp_init, _arg_init, main and _call_exit.**
- ❖ User can provide their own version of the above functions or use them as example codes to develop their own start-up code.
- ❖ The example project is on directory: demo/Startup

System Startup And Termination Diagram



➤ In crt0.S of Andes' version, some weak symbols are defined in order to customize your program easily depending on the features you need.

➤ These weak symbols are `__pre_c_init`, `__post_c_init`, `__cpp_init`, `__arg_init`, `main` and `__call_exit`.



IDE Related Features



- ❖ Fine-grained control of executable image generation
 - Start-up code control
 - Image loaded address and stack pointer assignment
 - Linker script control
- ❖ More control of tool chain utilities program
 - Objdump: displays information about one or more object files.
 - This information is mostly useful to programmers who are working on the compilation tools.
 - -x displays all available header information, including the symbol table and relocation entries;
 - -S displays source code intermixed with disassembly, if possible.
 - Readelf: displays information about one or more ELF format object files.
 - NM: lists the symbols from object files.
 - Objcopy: copies the contents of an object file to another. It is useful to generate the boot code binary image which will be burnt into FLASH chip.

Project Setting – Properties For C/C++ Build



Configuration Settings

Tool Settings | Toolchain | Build Settings

Andes C Compiler

- Preprocessor
- Symbols
- Directories
- Optimization
- Debugging
- Warnings
- Miscellaneous

Andes C Linker

- General
- Libraries
- Miscellaneous
- Shared Library Settings
- Loaded Address
- Linker Script

Andes Assembler

- General

Andes NM Tool

- Options

Andes Readelf Tool

- Options

Andes Objdump Tool

- Options

Andes Objcopy Tool

- Options

Do not use standard start files (-nostartfiles)

Do not use default libraries (-nodefaultlibs)

No startup or default libs (-nostdlib)

Omit all symbol information (-s)

☒ No shared libraries (-static)

☐ Relax branches on certain targets. (-mrelax)

☐ Write a map file. (-Map)

☐ Argc/Argv. (-mcr-arg=yes)

☐ Exit. (-mcr-exit=yes)

Program Start Address. 0x 500000

Data Address. 0x

Stack Address. 0x 4000000

Loaded Address






Use system default or choose either one.
With Linker Script precedes Loaded Address

Linker Script (-T) Browse...

Linker Script

Target Management (Local ICE and VEP)



Target	Run	Window	Help
	Fork VEP Target	Alt+V	
	Fork ICE Target	Alt+C	
	Fork REP Target	Alt+R	
	Query Target	Alt+Q	
	Input Target	Alt+I	

ICE Target Setting

Specify the launch options of ICE Target

☐ Enable Console setting
Console setting: enable AndeSight to take over the UART of Target

Port
Host's COM which connects to UART of Target (-x)

☐ Enable User Specify Options
Port setting: assign the port number used by GDBAgent

Port
Use specified TCP port (-p)

ICE Target Options

Board Name
Name of evaluation board (-name)

VEP Target Setting

Specify the launch options of VEP Target

VEP List

- ADP-AG101.vep
- ADP-XC5-N1033A-S.vep
- ADP-XC5-N1213-S.vep
- ADP-XC5-N1233-S.vep
- Andes-demo.vep
- Basic-1213H-30T2G.vep
- Basic-1233-S.vep
- Basic-1233H-30T2G.vep

Location:

☐ Enable User Specify Options
Port setting: assign the port number used by GDBAgent

Port
Use specified TCP port (-p)

VEP Target Options

Log file
Log VEP or ISS output to the specified file (-log)
[default is in {ANDESIGHT_INSTALL_DIR}\GDBAgent]

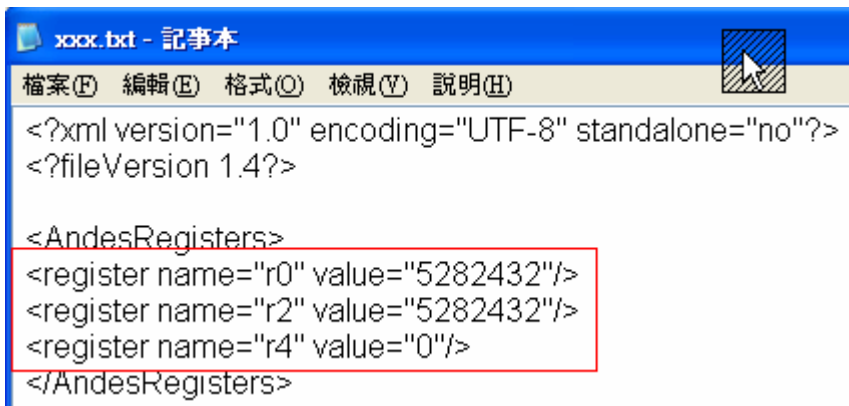
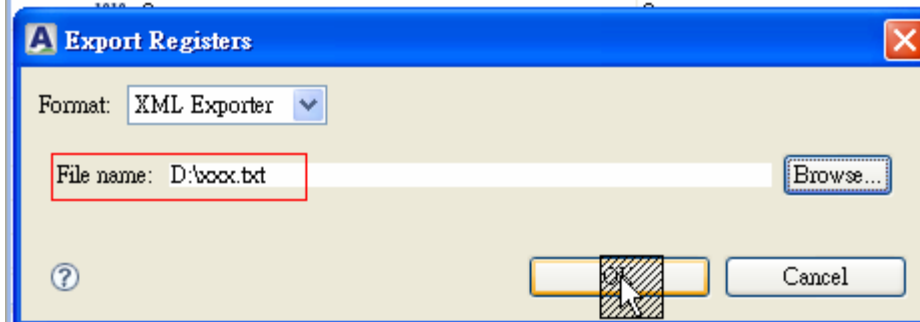
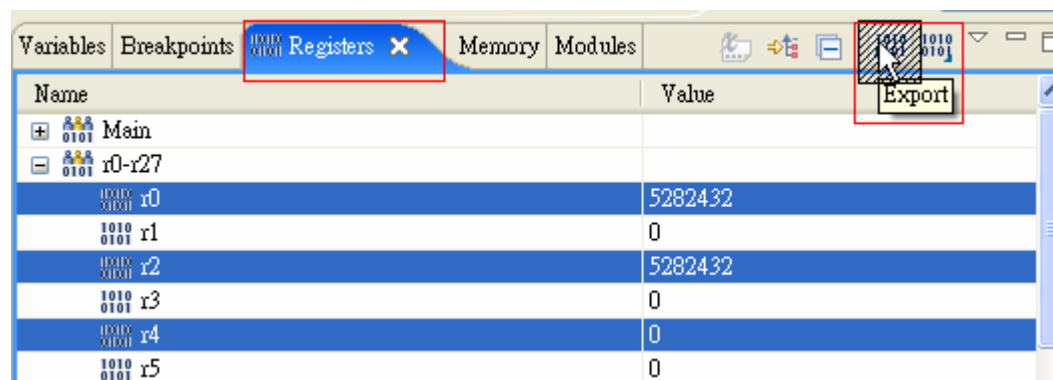
☒ Verbose console mode
Redirect sid's standard IO to GDBAgent's console (-verbose)

Debugging Assistant



- ❖ Save and Restore content of Register and Memory to file
- ❖ Support CPU and BUS view
- ❖ Hardware breakpoint setting
- ❖ Access SoC registers

Debugging Assistant - Save CPU Registers Content To File



Debugging Assistant – Restore CPU Registers Content From File



Variables	Breakpoints	Registers	Memory	Modules
Name			Value	
+	0101	Main		
-	0101	r0-r27		
	0101	r0	5282432	
	0101	r1	0	
	0101	r2	5282432	
	0101	r3	0	
	0101	r4	0	
	0101	r5	0	

Import Registers

Format: XML Importer

File name: D:\xxx.txt

Browse...

OK Cancel

xxx.txt - 記事本

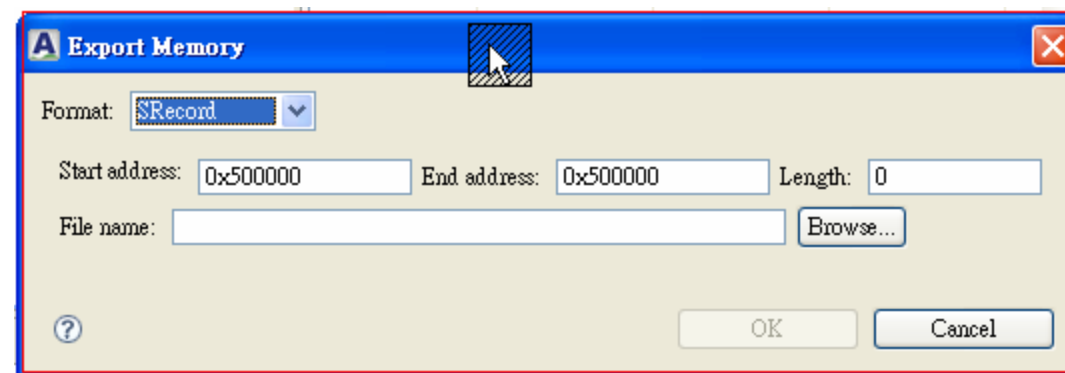
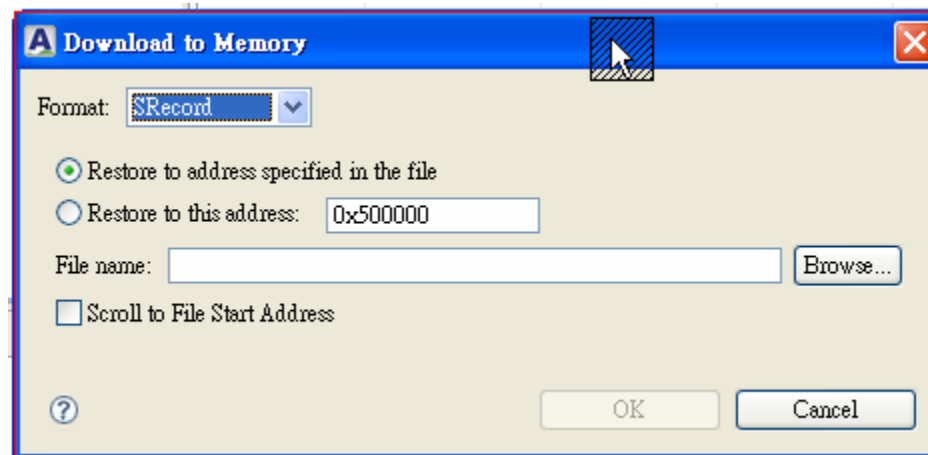
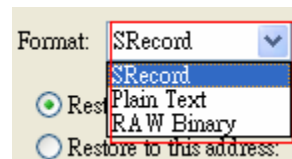
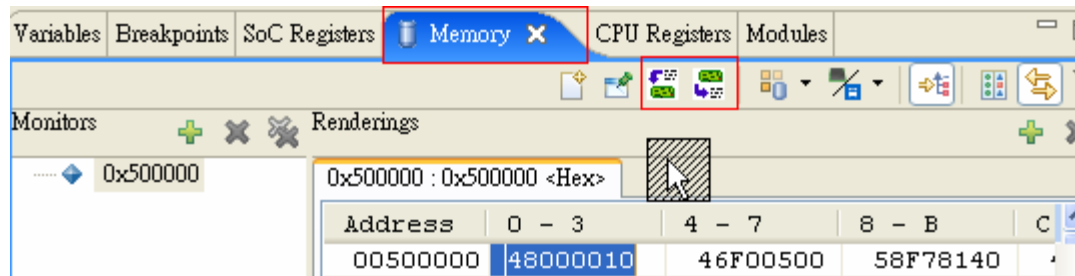
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<?fileVersion 1.4?>

<AndesRegisters>
<register name="r0" value="5282433"/>
<register name="r2" value="5282434"/>
<register name="r4" value="100"/>
</AndesRegisters>
```

+	0101	r0-r27		
	0101	r0	5282433	
	0101	r1	0	
	0101	r2	5282434	
	0101	r3	0	
	0101	r4	100	
	0101	r5	0	

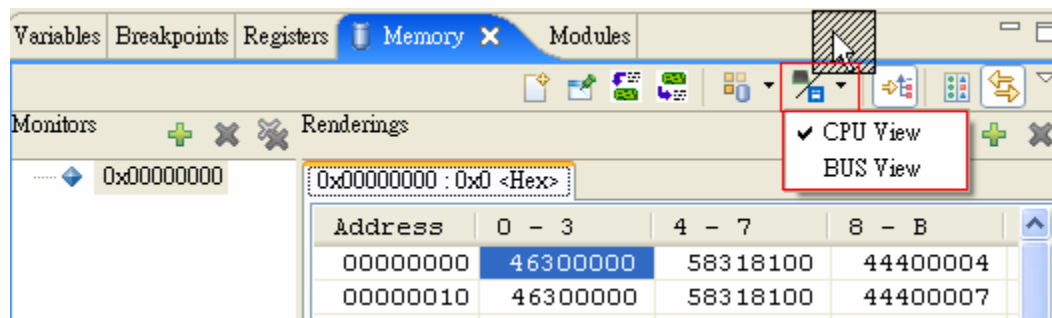
Debugging Assistant - Operation On Memory Content



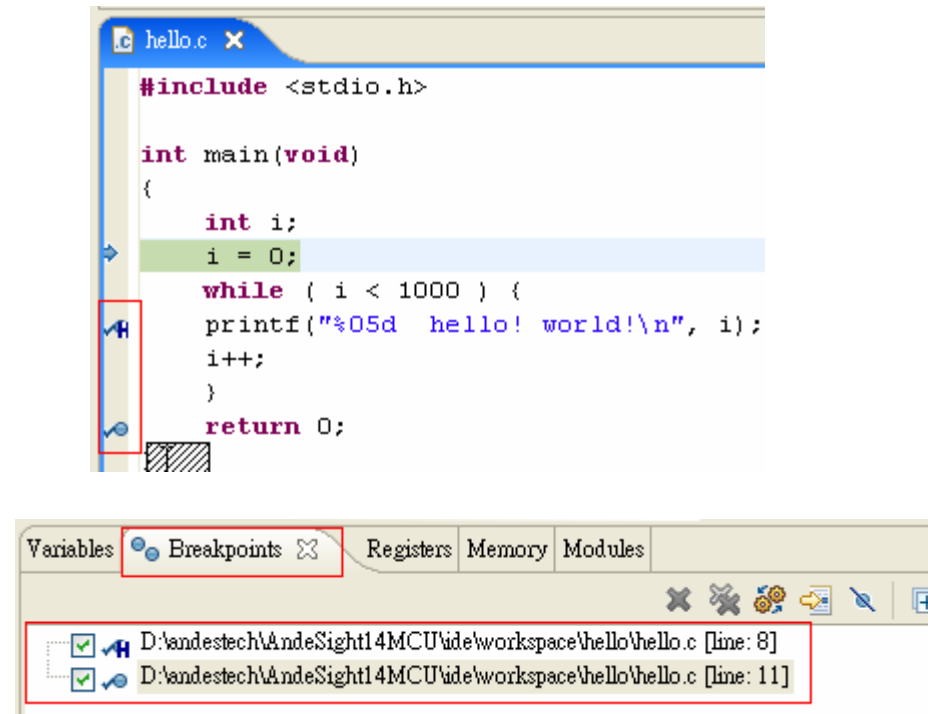
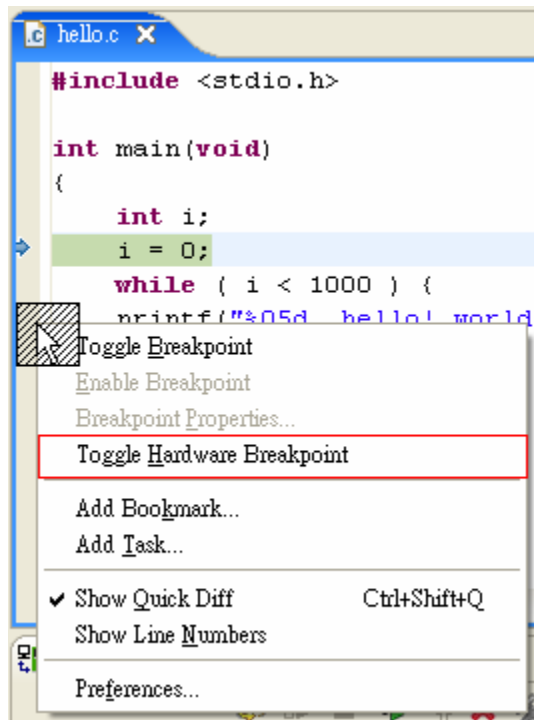
Debugging Assistant - CPU and BUS View



- ❖ When cache is enable, the memory content may differ from CPU or BUS view
- ❖ The CPU view always get the content from cache first
- ❖ The BUS view get the content from Memory directly



Debugging Assistant - Hardware Breakpoint Setting



When users set the SW break point (BP) on read only address, the debugger will detect and use HW BP instead. But be aware that the HW BP resources are limited.

Debugging Assistant - Access SoC Registers



❖ Format:

```
■ {  
  "groups": [  
    {  
      "GroupName": "Initial INTC and TIMER",  
      "registers": [  
        {  
          "registerName": "IRQMR",  
          "address": "0x98800004",  
          "readOnly": false,  
          "data": "0x0"  
        },  
        {  
          "registerName": "TmCR",  
          "address": "0x98400030",  
          "readOnly": false,  
          "data": "0x00000005"  
        }  
      ]  
    }  
  ]  
}
```

- Note: the value can be express as 0xffff – hex-decimal; 07777 – oct; 9999 - decimal





Debugging Assistant - SoC Registers View

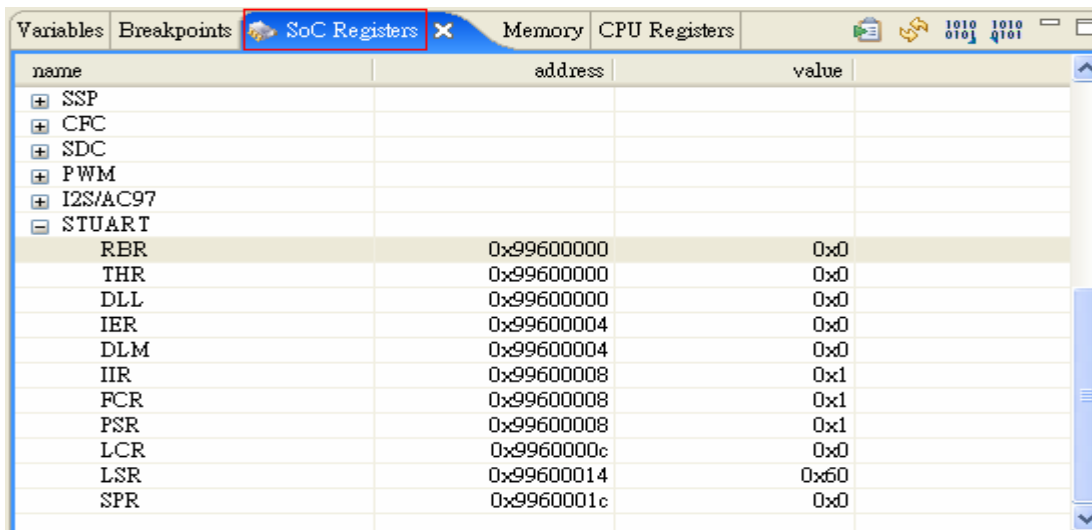


❖ Default SoC map: ~/SoC/AG101.regs

❖ Tool icons



-  Switch to Default View
-  Refresh All
-  Write Registers
-  Export Registers



name	address	value
+	SSP	
+	CFC	
+	SDC	
+	PWM	
+	I2S/AC97	
-	STUART	
	RBR	0x99600000 0x0
	THR	0x99600000 0x0
	DLL	0x99600000 0x0
	IER	0x99600004 0x0
	DLM	0x99600004 0x0
	IIR	0x99600008 0x1
	FCR	0x99600008 0x1
	PSR	0x99600008 0x1
	LCR	0x9960000c 0x0
	LSR	0x99600014 0x60
	SPR	0x9960001c 0x0

Utilities



❖ Flashing (In System Programming)

■ Target

• Target program based

- Download burner program to target system memory and execute this program to burn the image into target Flash chip

• Host script based

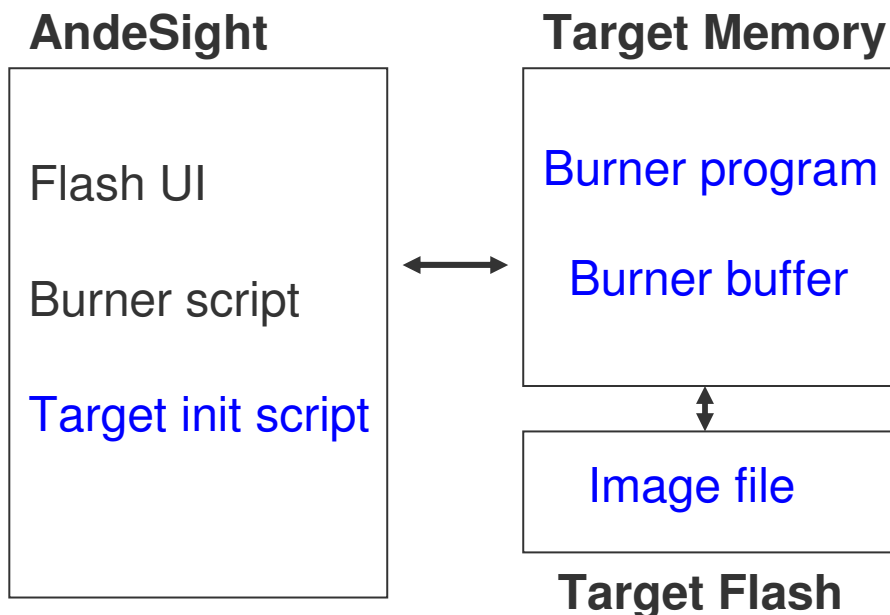
- The script program executes on host and burns the image into target Flash chip.

■ VEP

❖ RTOS-awareness (uC/OS-II)

- Use Script language for table creation, update and display (for OS awareness)
- Show the task (and its context), and event status

Flashing (In System Programming)



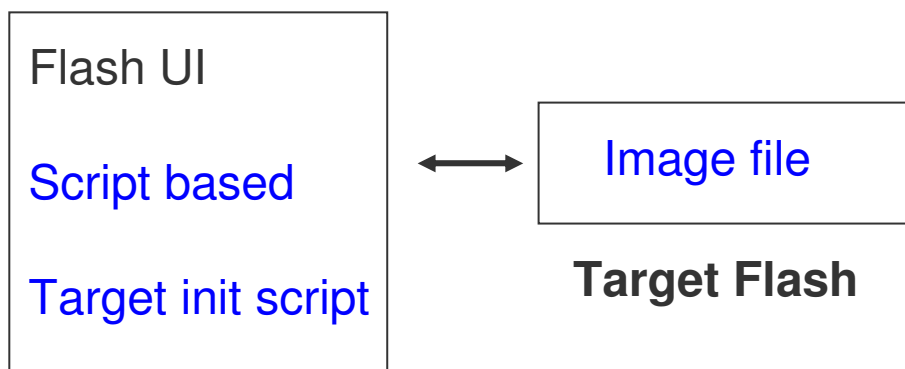
Burner program:

```
main( int bufaddr, int bufsize,
      int romaddr, int iobase,
      int binsize)
```

Burner:

1. must call `_buffer_done()` with error code when a block of buffer is done
2. Define a variable named `flash_block_size` to hold the block size of FLASH.

A dedicate `crt0.o`, provided by Andes, is used.

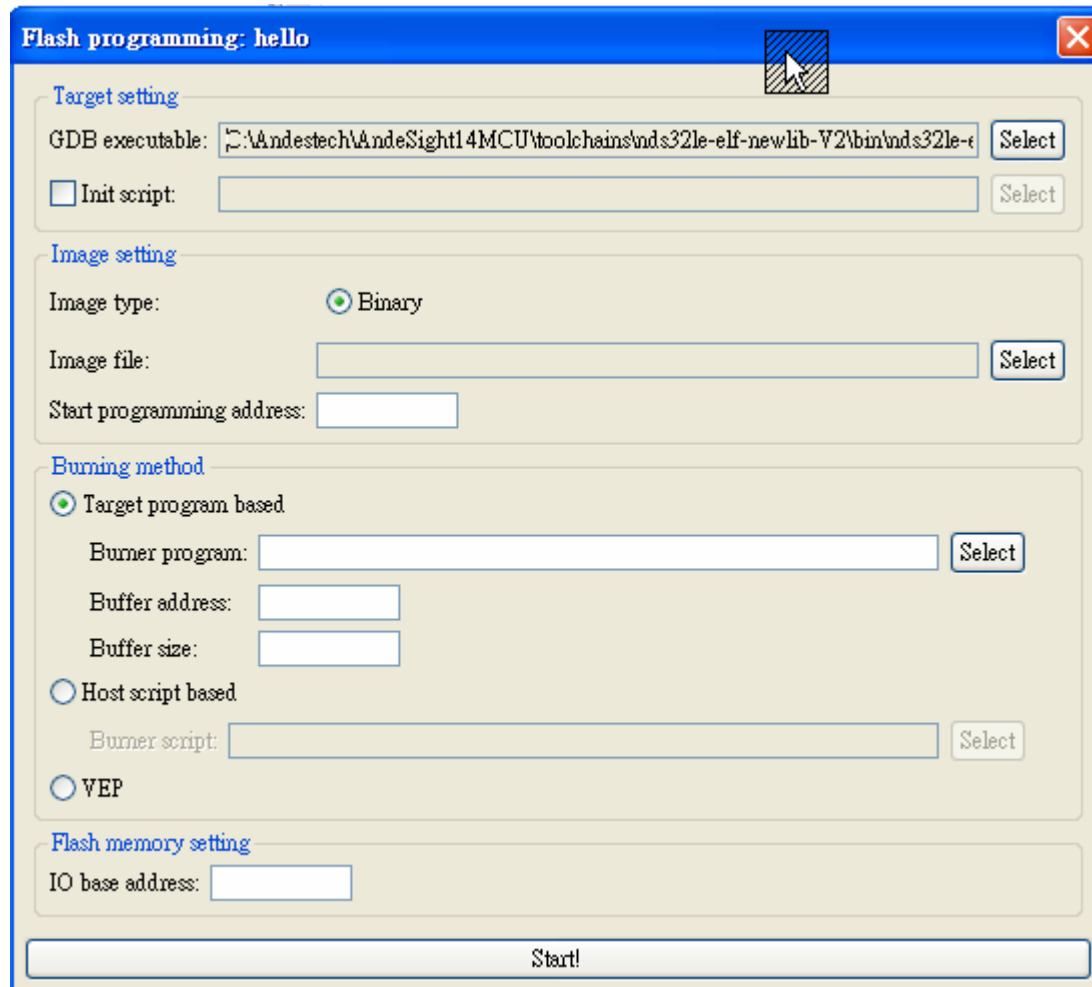
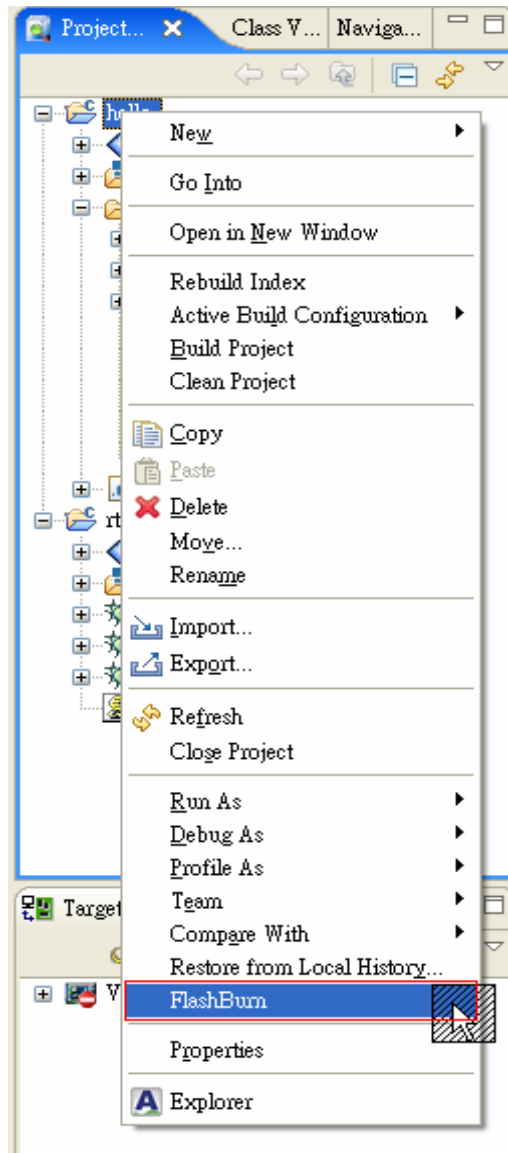


Script commands (gdb command file):

- `program_flash(`
 1. Target address
 2. Image file
 3. Image file size
 4. Flash IO base)
- `erase_flash`
- `flash_block_size`

AndeSight

Flashing (In System Programming)



RTOS-Awareness (uC/OS-II)



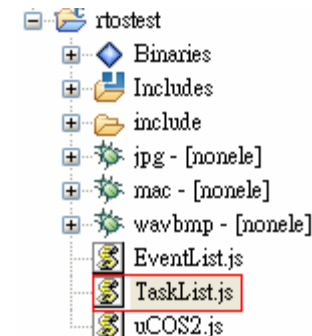
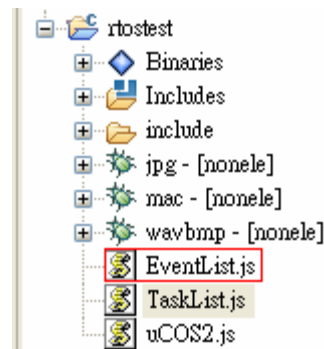
- ❖ Show the task (and its context), and event status
- ❖ The script is located under ~/demo/SampleScript/uCOS2
- ❖ Task:
 - Status fields:
 - "magic", "id", "prio", "state", "delay", "name", "stack ptr", "stack bottom", "stack size", "event", "msg", "current", "options"
 - State of a task can be "READY", "SEM", "MBOX", "QUEUE", "SUSPEND", "MUTEX", "MULTI", "UNKNOWN"
 - Context
 - Registers: \$r0~\$r31, \$d0, \$d1, \$ipc, \$ipsw.
- ❖ Event status
 - "magic", "type", "ptr", "cnt", "waiting", "name", "waiting task prio"
 - Type of an event can be "UNUSED", "MBOX", "QUEUE", "SEM", "MUTEX", "FLAG", "TMR"

RTOS-Awareness (uC/OS-II)



Name	Value
ipsw	0x00000008
ipc	0x00501400
d1lo	0x20202020
d1hi	0x21212121
d0lo	0x22222222
d0hi	0x23232323
r0	0x00000008
r1	0x00000007
r2	0x0056bc8
r3	0x0051d000
r4	0x0056e3dc
r5	0x0051d000
r6	0x00000009
r7	0x0200eff4
r8	0x00000001
r9	0x0056e850
r10	0x0a0a0a0a
r11	0x0b0b0b0b
r12	0x0c0c0c0c
r13	0x005076ac
r14	0x0056f3d2
r15	0x005054da
r16	0x0056f000
r17	0x11111111
r18	0x12121212
r19	0x13131313
r20	0x14141414
r21	0x15151515
r22	0x16161616
r23	0x17171717
r24	0x18181818
r25	0x19191919
r26	0x1a1a1a1a
r27	0x1b1b1b1b
fp	0x00506de4
gp	0x1d1d1d1d
lp	0x00501400
sp	0x0200eec8

Magic	Type	Ptr	Cnt	Tasks	Waiting	Name
0x56e7e4	SEM	0x00000000	1	0	0	OS-TmrLock
0x56e808	SEM	0x00000000	0	1	1	OS-TmrSig
0x56e82c	MBOX	0x00000000	0	1	1	?
0x56e850	SEM	0x00000000	0	1	1	?



Magic	Id	Priority	State	Delay	Name	Stack ptr	Stack bottom	Stack size	Event	Msg	Current	Opt
0x0056ec80	0	17	SEM	0	Ethif BH	0x0200eec8	0x00000000	0x00000000	0x0056e850	0x00000000	0	NONE
0x0056ec24	0	14	MBOX	3	tcpip_thread	0x02001e98	0x00000000	0x00000000	0x0056e82c	0x00000000	0	NONE
0x0056ebc8	0	30	READY	1	Main Task	0x0052d35c	0x00000000	0x00000000	0x00000000	0x00000000	0	NONE
0x0056eb6c	65533	5	SEM	0	uC/OS-II Tmr	0x0056e4cc	0x0056e3e4	0x00000080	0x0056e808	0x00000000	0	STK_CHK,STK_CLR
0x0056eb10	65534	62	READY	18	uC/OS-II Stat	0x0056e284	0x0056e17c	0x00000080	0x00000000	0x00000000	0	STK_CHK,STK_CLR
0x0056eab4	65535	63	READY	0	uC/OS-II Idle	0x0056e72c	0x0056e5e4	0x00000080	0x00000000	0x00000000	1	STK_CHK,STK_CLR



Thank You