



Работа с ЦСП TMS320C674x: инструментальные средства



Средства разработки программного обеспечения ЦСП TMS320C67x

Системы моделирования
(Matlab, LabView), языки
высокого уровня (Си)



Программа-симулятор



Стартовые наборы
разработчика и оценочные
платы



Эмуляторы





Программирование ЦСП TMS320C67x

Типовой процесс разработки ПО для ЦСП:

1. Разрабатывается алгоритм обработки сигнала, проводится его моделирование и исследование в системах MATLAB, LabView. Возможно создание и отладка ПО на языках высокого уровня (обычно Си), реализующее требуемый алгоритм.
2. В специальной среде программирования ЦСП формируется проект, нацеленный на выполнение на конкретном сигнальном процессоре. Проект может разрабатываться на основе языка Си или в кодах ассемблера. Разработка проекта может производиться «вручную» или выполняться автоматически средствами MATLAB или LabView. Моделируется выполнение ПО на ЦСП. Моделировать работу процессора программно, не имея самого процессора, позволяют программы-симуляторы. Программа-симулятор позволяет осуществить отладку, оптимизацию и тестирование программного обеспечения, оценить параметры исполнения ПО на данном ЦСП.



3. Разработанное и отлаженное в симуляторе программное обеспечение тестируется на реальном, «живом» процессоре. Для этого используют **проверочные модули**. Это платы, включающие в свой состав сигнальный процессор и типовые элементы систем ЦОС - дополнительную памяти, АЦП/ЦАП и другие. На таких модулях можно полностью реализовать простые цифровые системы обработки аналогового сигнала, а порой и получить законченный продукт, не разрабатывая собственного аппаратного обеспечения. Проверочные модули можно разделить на **стартовые наборы** и **отладочные модули**. Отладочные модули отличаются от стартовых наборов большим числом специализированных аппаратных узлов, более высоким их качеством, ценой.

4. Заключительным этапом разработки является отладка ПО на сигнальном процессоре, встроенном в конечное оборудование. При этом сохраняется возможность доступа к процессору с персонального компьютера для анализа его работы и отладки в реальном времени. Такую возможность обеспечивают устройства сопряжения ПК и ЦСП - **эмуляторы**.



Среда Code Composer Studio

Позволяет:

Создавать программы
(средства редактирования, связь
с Матлаб)

Строить проект (компилятор
C/C++, оптимизатор)

Загружать код на целевое
оборудование (разные
платформы)

Анализировать и управлять
процессом выполнения
программы (средства
визуализации, в том числе
обмен данными в реальном
времени)

The screenshot displays the Code Composer Studio (CCS) interface for a C54X simulator. The main window shows assembly code for a program named 'lfm11-1.pjt'. The code includes instructions like 'SSEX INTM', 'RSEX OVM', 'RSEX SXM', 'RSEX CPL', 'RSEX XF', 'RSEX C16', 'STM 0h, STO', 'STM 4028h, PMST', 'STM 1000h, BSCR', 'STM 100h, SP', 'STM 0h, _edata', 'STM 0ffffh, IFR', 'STM 7ffffh, SWWSR', 'PLLX1', 'STM 0h, CLKMD', 'BITF CLKMD, 1h', 'BC PLLX1, TC', 'PLLXN', 'STM 70e6h, CLKMD', 'BITF CLKMD, 1h', 'BC PLLXN, NTC', 'STM 80h, AR0', 'NOP', 'RPT2 A, 7fh', 'STL A, *AR0+', 'STM 3981h, AR0', 'NOP', 'RPT2 A, 7fh', 'STL A, *AR0+', 'LD #0h, DP', 'CALL BUF_INIT', 'CALL SMCBSP_rout.asm:39:6185', 'STM 1fh, TCR', and 'STM 0ed5ah, PRD'. The right side of the window shows a memory dump with values like '0x0000FE0C', '0x0000', '0x0000', etc. The bottom left shows a build log with the message 'Build Complete, 0 Errors, 0 Warnings, 0 Remarks.' The bottom right shows a register window with values for PC, TRN, SP, STO, AR0, A, and AR1.



Среда Code Composer Studio

Вся работа с ЦСП TMS320C67x, включая этапы разработки и отладки ПО, симуляции, отладки на «живом» процессоре и работы с конечным оборудованием, ведется через персональный компьютер с использованием программной среды разработки и отладки программного обеспечения Code Composer Studio.

Code Composer Studio позволяет:

создавать программы - писать и редактировать тексты программ на Си и ассемблере, импортировать проект для ЦСП из систем MATLAB или LabView.

осуществлять построение проекта - генерировать на основе написанных программ машинных кодов для заданного ЦСП. При этом доступны компилятор языков Си/Си++ и средства оптимизации Си и ассемблерных кодов.



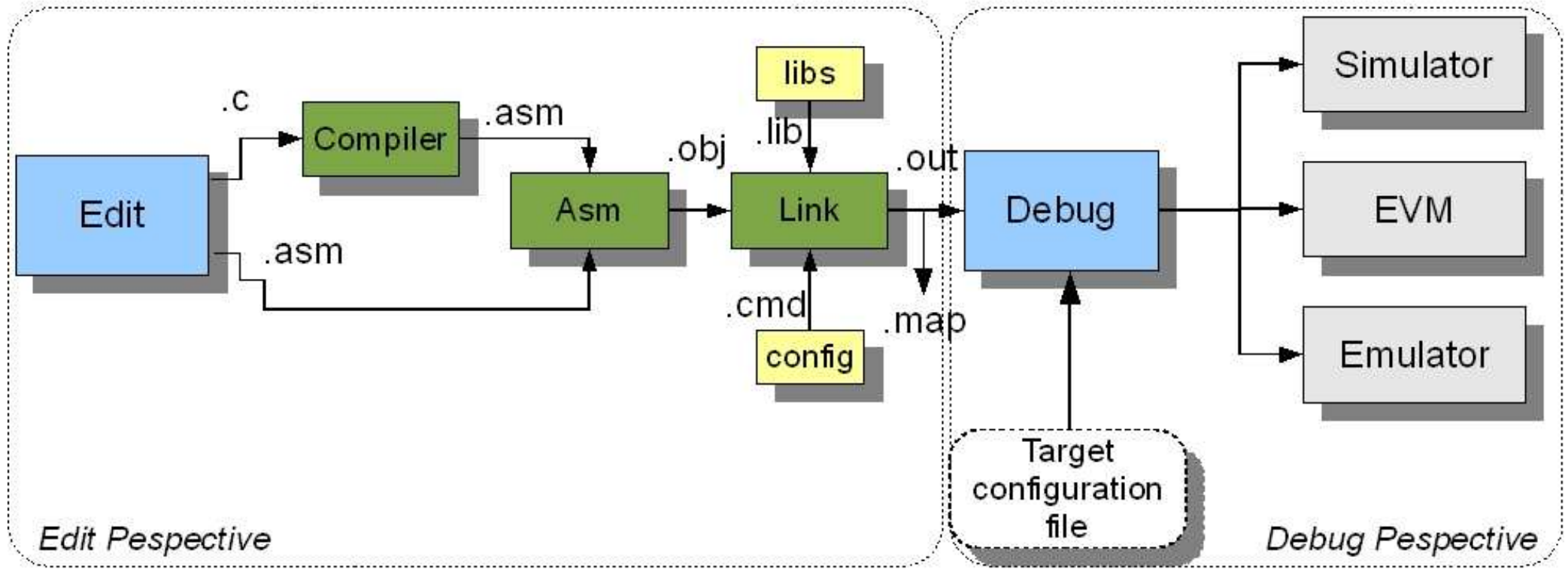
Загружать код на целевое оборудование - инициировать выполнение кодов программ в режиме симуляции или при их наличии на различных аппаратных платформах (проверочных модулях, платах собственного изготовления).

Анализировать и управлять процессом выполнения программы - наблюдать за состоянием внутренних ресурсов процессора и изменять их состояние, что необходимо для отладки ПО. Имеются удобные средства визуализации состояния процессора (построение графиков, анализ спектра и др.). Поддерживается полноценный анализ и управление состоянием процессора при программировании аппаратного обеспечения и в реальном времени.



Среда Code Composer Studio v6

Code Composer Studio v6





CCS Edit - omp_fir/omp_DOTP.c - Code Composer Studio

File Edit View Navigate Project Run Scripts Window Help

Quick Access CCS Edit CCS Debug

Project Explorer

- 674x tmp example
- afir
- firstBIOS
- firstIO
- Lab5
- omp_fir [Active - Debug]**
 - Binaries
 - Includes
 - Debug
 - data.h
 - omp_DOTP.c
 - omp_FIR.c
 - omp_hello.c
 - dotp_solution.txt
 - omp_config.cfg
- openEM_lab
- UAIttry
- UIA example

```
34
35 // #pragma DATA_SECTION(x, ".12data")
36 // #pragma DATA_ALIGN(x, 8)
37 short x[COUNT] = {X_ARRAY};
38 volatile int y1 = 0;
39 volatile int y2 = 0;
40
41 #define NTHREADS 4
42
43 void main()
44 {
45
46 int nthreads, tid;
47 int acc, i;
48 Int32 t0, t1, t2, t3, t4, t5, t6, t7, tt0, tt1, tt2, tt3, tt4, tt5, tt6, tt7;
49
50 nthreads = NTHREADSc;
51
52 omp_set_num_threads(NTHREADS);
53
54 // DOTP sequential code
55
56 t0 = (Int32)Timestamp_get32();
57
58 acc = 0;
59 for (i = 0; i < COUNT; i++)
60     acc += a[i]*x[i];
61
62 y1 = acc;
63
64 t1 = (Int32)Timestamp_get32();
65
66 tt0 = t1 - t0;
67
68 #pragma omp parallel private(tid, i) //reduction(+:acc)
69 {
70     tid = omp_get_thread_num();
71 }
```

Outline Target..

type filter text

- Projects
- User Defined
 - EVM6678.ccxml
 - NewTargetConfiguration.ccxml

Console

CDT Build Console [omp_fir]

```
>> Compilation failure
gmake: *** [omp_DOTP.obj] Error 1
gmake: Target 'all' not remade because of errors.

**** Build Finished ****
```

Problems Advice

1 error, 8 warnings, 0 others

Description	Resource	Path	Location	Typ
Errors (1 item)				
#20 identifier "NTHREADSc" is undefined	omp_DOTP.c	/omp_fir	line 50	C/C-
Warnings (8 items)				
#179-D variable "t5" was declared but never referenced	omp_DOTP.c	/omp_fir	line 48	C/C-
#179-D variable "t6" was declared but never referenced	omp_DOTP.c	/omp_fir	line 48	C/C-
#179-D variable "t7" was declared but never referenced	omp_DOTP.c	/omp_fir	line 48	C/C-

Free License

- Workspace
- Project
- Perspective
- New File
- Project Options
- Build
- All windows
- Menu
- Target Config File



CCS Debug - firstBIOS/main.c - Code Composer Studio

File Edit View Project Tools Run Scripts Window Help

Quick Access | CCS Edit | CCS Debug

Debug | EVM6678.ccxml [Code Composer Studio - Device Debugging]

Group 1

- Blackhawk XD5560v2-USB Mezzanine Emulator_0(C66xx_0 (Suspended - SW Breakpoint))
- main() at main.c:67 0x0080C218
- _c_int00() at boot.c:180 0x0080D844 (the entry point was reached)
- Blackhawk XD5560v2-USB Mezzanine Emulator_0(C66xx_1 (Disconnected : Unknown))
- Blackhawk XD5560v2-USB Mezzanine Emulator_0(C66xx_2 (Disconnected : Unknown))
- Blackhawk XD5560v2-USB Mezzanine Emulator_0(C66xx_3 (Disconnected : Unknown))
- Blackhawk XD5560v2-USB Mezzanine Emulator_0(C66xx_4 (Disconnected : Unknown))
- Blackhawk XD5560v2-USB Mezzanine Emulator_0(C66xx_5 (Disconnected : Unknown))
- Blackhawk XD5560v2-USB Mezzanine Emulator_0(C66xx_6 (Disconnected : Unknown))

Debug | Registers | Expressions | Breakpoints

Name	Value	Description
SP	0x00814890	Core
FP	0x00814890	Core
A0	0x00000001	Core
A1	0x00000001	Core
A2	0x00000000	Core
A3	0x00000000	Core
A4	0x00000000	Core
A5	0xFFFFFFFF	Core

Disassembly | Enter location here | \$CDW\$taskFxn1\$2\$B, main():

```
0080c218: 01BC94F6 STW.D2T2 B3,*B15--
0080c21c: E040000C .fthead n, l, W,
71 System_printf("enter main()\n");
0080c220: 021B482A MVKH.S2 0x3690,B4
0080c224: 020040EA MVKH.S2 0x810000,
0080c228: 10076C13 CALLP.S2 xdc_runtf
0080c22c: 023C22F6 || STW.D2T2 B4,*+B15[
73 BIOS_start(); /* does not return */
0080c230: 1007B812 CALLP.S2 ti_sysbic
74 return(0);
$CDW$15.
```

Memory Browser | 0x00800000

32-Bit Floating Point | Memory Rendering 1>

Address	Value
0x00800000	CSL_EMIF4F_GetSDRAMConfig
0x00800008	2.873869e-34
0x00800010	1.06163e-37
0x00800018	-1.163341e-08
0x00800020	-4.954799e+07
0x00800028	6.258309e+07
0x00800030	-4.958024e+07
0x00800038	5.160872e-38
0x00800040	5.169912e-38
0x00800048	6.258309e+07
0x00800050	-4.961301e+07
0x00800058	-6.1897e+27
0x00800060	5.183975e-38
0x00800068	6.258309e+07
0x00800070	7.110193e-33
0x00800078	1.151205e+27
0x00800080	5.426563e-38
0x00800088	-2.321232e+26
0x00800090	5.160872e-38
0x00800098	6.258309e+07
0x008000A0	-4.961301e+07
0x008000A8	-6.1897e+27
0x008000B0	5.183975e-38
0x008000B8	6.258309e+07
0x008000C0	7.110193e-33

Console | EVM6678.ccxml

```
EVM6678.ccxml
C66xx_0: GEL Output: Enabling LPU ...Done
C66xx_0: GEL Output: Configuring CPSW ...
C66xx_0: GEL Output: Configuring CPSW ...Done
```

main.c | omp_FIR.c | omp_DOTP.c | omp_hello.c | EVM6678.ccxml

```
46 Task_sleep(10);
47
48 while(1)
49 {
50
51 platform_led(2, PLATFORM_LED_ON, PLATFORM_USER_LED_CLASS);
52 platform_delay(2000); //Task_sleep(2000); //platform_delay(50000);
53 platform_led(2, PLATFORM_LED_OFF, PLATFORM_USER_LED_CLASS);
54 platform_delay(2000); //Task_sleep(2000); //platform_delay(50000);
55
56 // Log_info1("Hello from task 1! [%u]", 555);
57
58 }
59
60 System_printf("exit taskFxn()\n");
61
62
63 /*
64 * ===== main =====
65 */
66 int main()
67 {
68 /*
69 * use ROW->SysMin to view the characters in the circular buffer
70 */
71 System_printf("enter main()\n");
72
73 BIOS_start(); /* does not return */
74 return(0);
75 }
76
```



CCS Edit - firstBIOS/main.c - Code Composer Studio

File Edit View Navigate Project Run Scripts Window Help

Project Explorer: firstBIOS [Active - Debug]

- 674x tmp example
- afir
- firstBIOS
- Labs
- omp_fir
- Includes
- Debug
- data.h
- omp_DOTP.c
- omp_FIR.c
- omp_hello.c
- dotp_solution.txt
- omp_config.cfg
- openEM_lab
- UATry
- UJA example

```
main.c  
40 }  
41  
42 void taskFxn1  
43 (  
44     System_pr  
45     Task_sLee  
46     while (1)  
47     (  
48         platform  
49         platform  
50         platform  
51         platform  
52         platform  
53         platform  
54         platform  
55         Log_inf  
56 //  
57     )  
58 }  
59  
60 System_pr  
61 }  
62  
63 /*  
64 * =====  
65 */  
66 int main ()  
67 {  
68     /* use RC  
69     */  
70     System_pr  
71     BIOS_star  
72     return (0)  
73 }  
74  
75 }  
76
```

New CCS Project

Create a new CCS Project.

Target: <select or type filter text> Generic C674x Device

Connection: C67XX [C6000]

Project name: first_proj

Use default location:

Location: E:\W\yasev\SIP3\soft\workspace_s\first_proj

Compiler version: TI v7.4.14

More...
Advanced settings
Project templates and examples

type filter text

- Empty Projects
- Empty Project
- Empty Project (with main.c)
- Empty Assembly-only Project
- Empty RTSC Project
- Basic Examples
- Hello World
- IPC and I/O Examples
- OMP Examples
- SYS/BIOS
- System Analyzer (UJA)
- TI-RTOS for C6000

Creates an empty project fully initialized for the selected device. The project will contain an empty main.c source-file.

< Back Next > Finish Cancel

Console

EVM6678.ccxml

```
0000x_0: GEL Output: Initializing LDC .....  
C66xx_0: GEL Output: Configuring CPSW ...  
C66xx_0: GEL Output: Configuring CPSW .... Done  
C66xx_0: GEL Output: Global Default Setup...  
Done.  
C66xx_0: GEL Output: Invalidate All Cache...  
C66xx_0: GEL Output: Invalidate All Cache...  
Done.  
C66xx_0: GEL Output: GEL Reset...
```

Warnings (3 Items)

Description	Resource	Path	Location	Type
#10247-D creating output section "platform_lib" without a SEC firstBIOS	main.c	/firstBIOS	line 39	C/C++
#112-D statement is unreachable	main.c	/firstBIOS	line 60	C/C++
#112-D statement is unreachable				C/C++

0 errors, 3 warnings, 0 others

Free License LE

15:31



CCS Edit - firstBIOS/main.c - Code Composer Studio

File Edit View Navigate Project Run Scripts Window Help

Project Explorer

- 674x tmp example
- afir
- firstBIOS [Active - Debug]
 - firstIO
 - Labs
 - omp_fir
 - Binaries
 - Includes
 - Debug
 - data.h
 - omp_DOTP.c
 - omp_FIR.c
 - omp_hello.c
 - dotp_solution.txt
 - omp_config.cfg
 - openEM_lab
 - UARTy
 - UIA example

main.c

```
39 }
40 }
41
42 void taskFxn1
43 (
44     System_pr
45     Task_slee
46     while (1)
47     (
48         (
49         platform
50         platform
51         platform
52         platform
53         platform
54         platform
55         Log_inf
56 //
57     )
58     )
59     System_pr
60     )
61 }
62
63 /*
64 * =====
65 */
66 int main ()
67 {
68     /*
69     * use RC
70     */
71     System_pr
72
73     BIOS_star
74     return (0)
75 }
76
```

Console

```
EVM6678.ccxml
C66xx_0: GEL Output: Unloading LDC .....Done
C66xx_0: GEL Output: Configuring CPSW ...
C66xx_0: GEL Output: Configuring CPSW ... Done
C66xx_0: GEL Output: Configuring CPSW ... Done
C66xx_0: GEL Output: Global Default Setup...
Done.
C66xx_0: GEL Output: Invalidate All Cache...
C66xx_0: GEL Output: Invalidate All Cache...
Done.
C66xx_0: GEL Output: GEL Reset...
```

Warnings (3 items)

- #10247-D creating output section "platform_lib" without a SEC firstBIOS main.c
- #112-D statement is unreachable main.c
- #112-D statement is unreachable main.c

0 errors, 3 warnings, 0 others

Project Explorer

- 674x tmp example
- afir
- firstBIOS [Active - Debug]
 - firstIO
 - Labs
 - omp_fir
 - Binaries
 - Includes
 - Debug
 - data.h
 - omp_DOTP.c
 - omp_FIR.c
 - omp_hello.c
 - dotp_solution.txt
 - omp_config.cfg
 - openEM_lab
 - UARTy
 - UIA example

main.c

```
39 }
40 }
41
42 void taskFxn1
43 (
44     System_pr
45     Task_slee
46     while (1)
47     (
48         (
49         platform
50         platform
51         platform
52         platform
53         platform
54         platform
55         Log_inf
56 //
57     )
58     )
59     System_pr
60     )
61 }
62
63 /*
64 * =====
65 */
66 int main ()
67 {
68     /*
69     * use RC
70     */
71     System_pr
72
73     BIOS_star
74     return (0)
75 }
76
```

Console

```
EVM6678.ccxml
C66xx_0: GEL Output: Unloading LDC .....Done
C66xx_0: GEL Output: Configuring CPSW ...
C66xx_0: GEL Output: Configuring CPSW ... Done
C66xx_0: GEL Output: Configuring CPSW ... Done
C66xx_0: GEL Output: Global Default Setup...
Done.
C66xx_0: GEL Output: Invalidate All Cache...
C66xx_0: GEL Output: Invalidate All Cache...
Done.
C66xx_0: GEL Output: GEL Reset...
```

Warnings (3 items)

- #10247-D creating output section "platform_lib" without a SEC firstBIOS main.c
- #112-D statement is unreachable main.c
- #112-D statement is unreachable main.c

0 errors, 3 warnings, 0 others

New CCS Project

Create a new CCS Project.

Target: <select or type filter text> Generic C674x Device

Connection: C67XX [C6000]

Project name: first_proj

Use default location

Location: E:\W\razev\SDSP\soft\workspace_s\first_proj

Compiler version: TI v7.4.14

Advanced settings

Output type: Executable

Output format: legacy COFF

Device endianess: little

Linker command file: <none>

Runtime support library: <automatic>

Project templates and examples

Warnings (3 items)

- #10247-D creating output section "platform_lib" without a SEC firstBIOS main.c
- #112-D statement is unreachable main.c
- #112-D statement is unreachable main.c

0 errors, 3 warnings, 0 others

Free License