

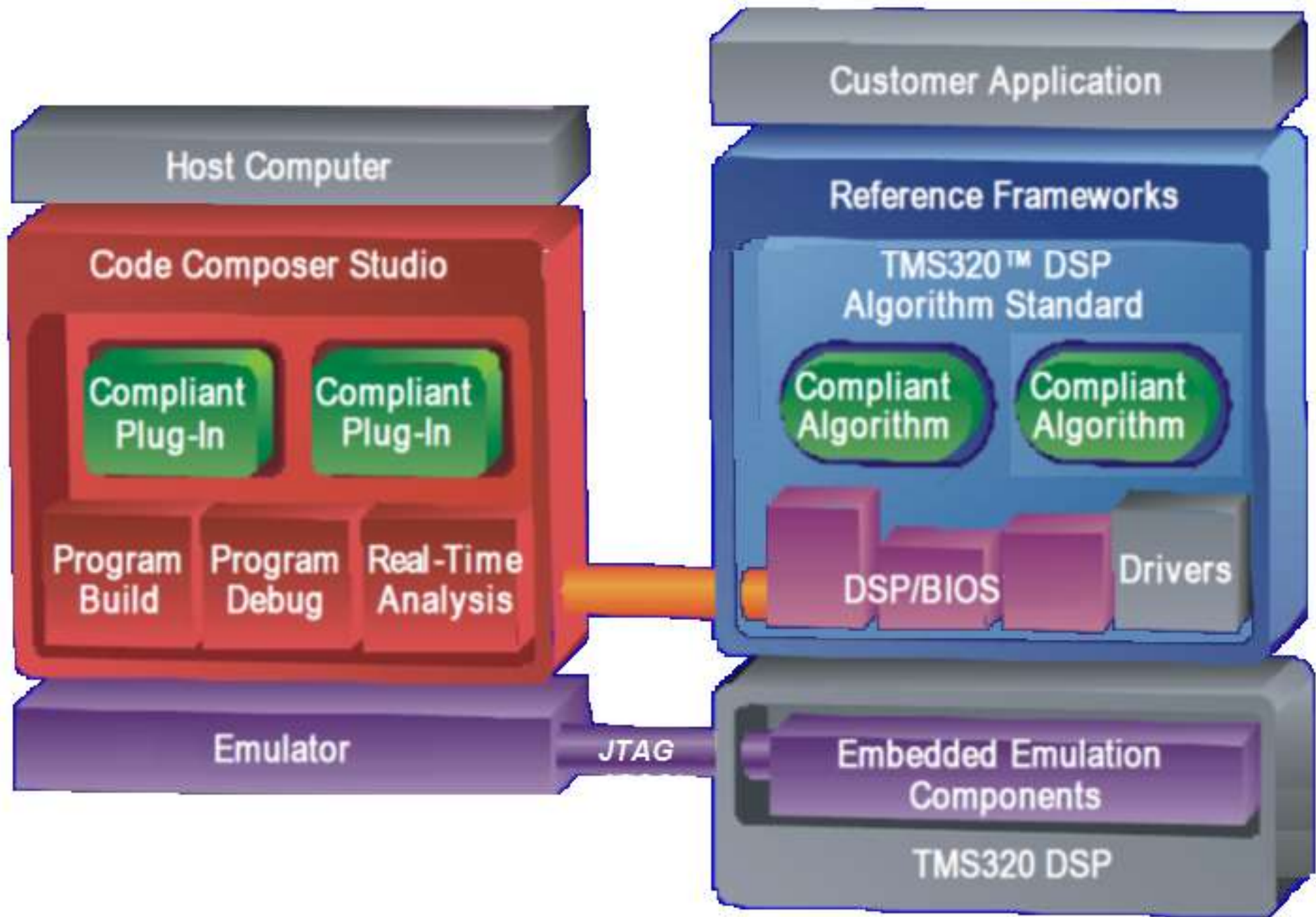


# Микропроцессорные устройства обработки сигналов

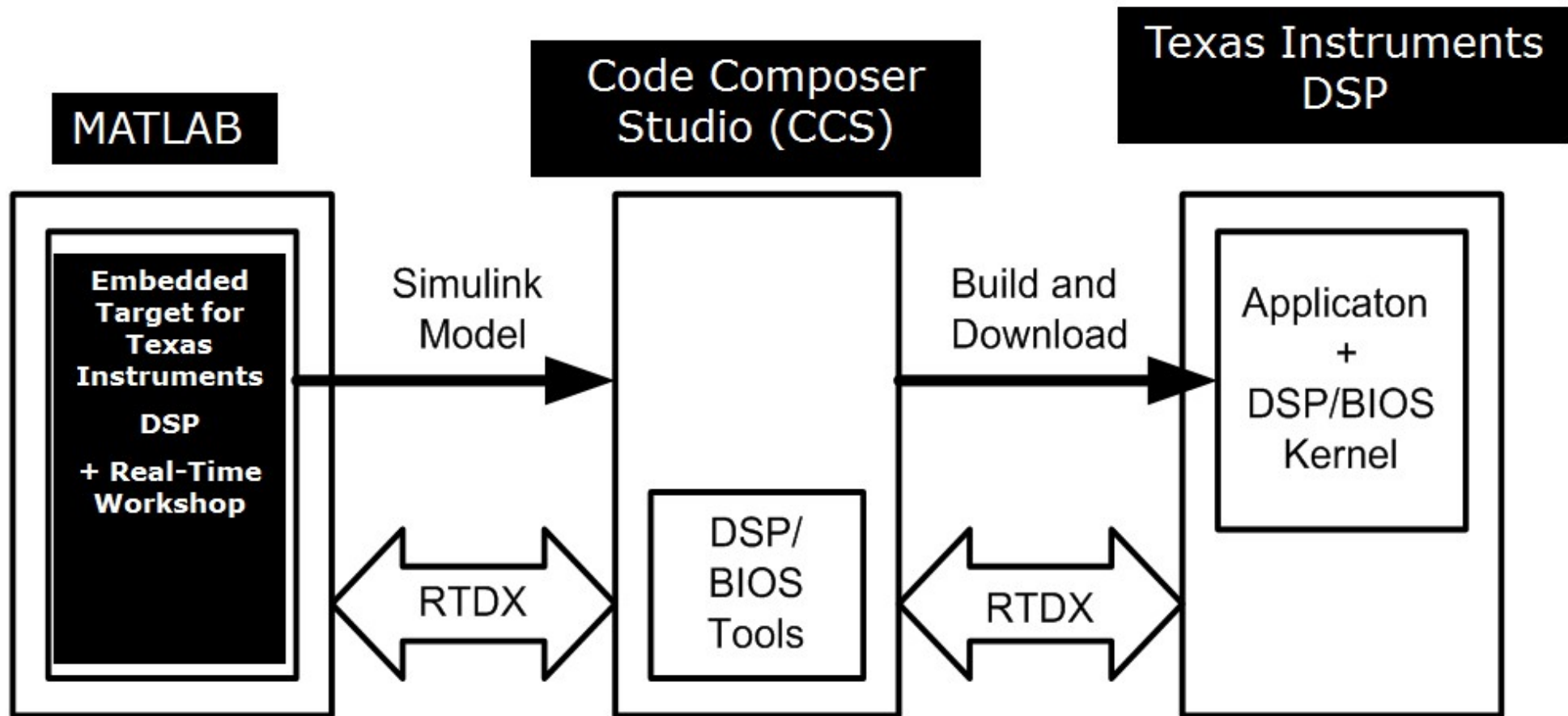
Лекция L17  
«Операционная система»

<http://vykhovanets.ru/course67/>

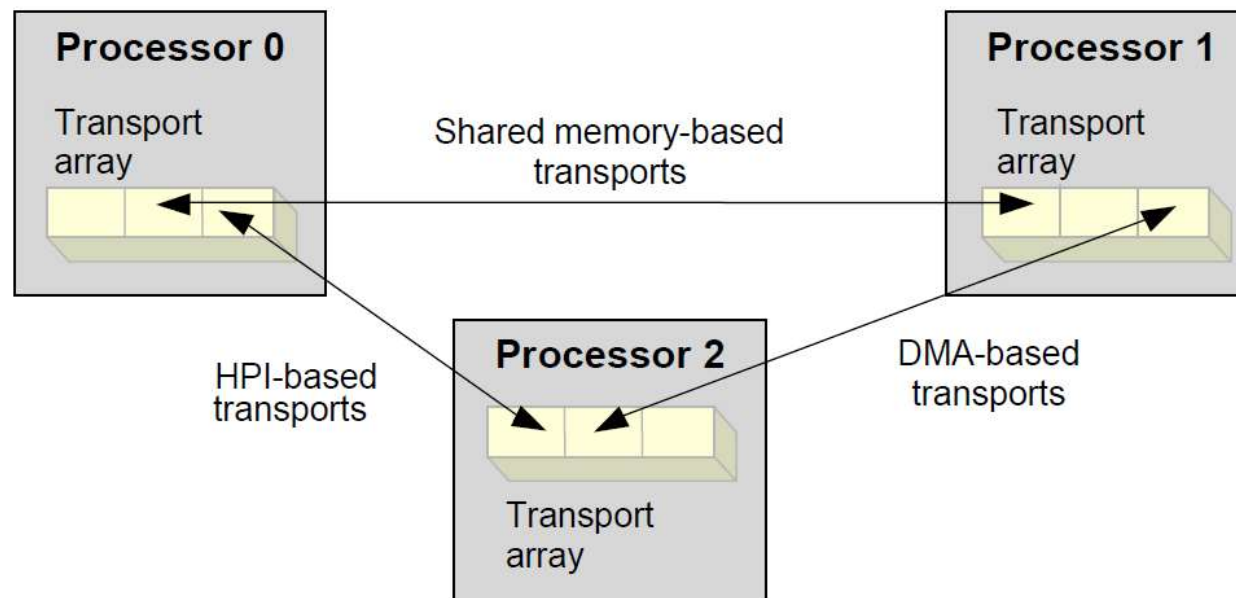
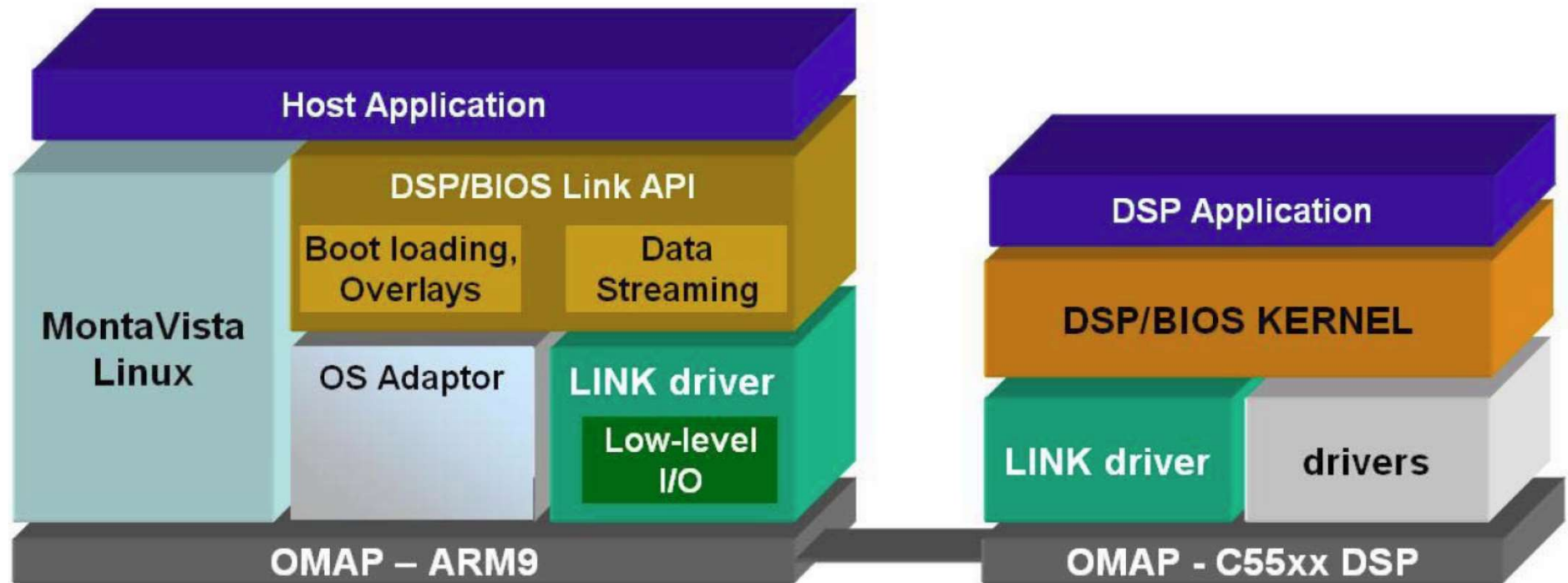
# Технология разработки



# Разработка алгоритмов



# Мультипроцессорная обработка



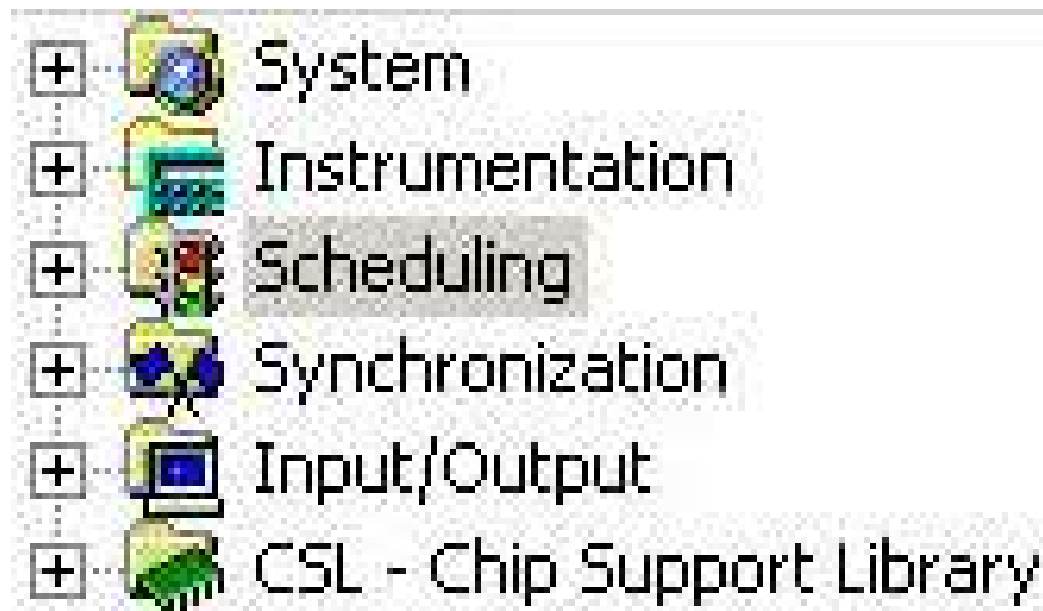
# DSP/BIOS

**DSP/BIOS** (Digital Signal Processing Base Instrumentation Operation System) – операционная система реального времени, предоставляющая следующие сервисы:

- мультизадачный планировщик задач;
- аппаратная абстракция устройств ввода-вывода данных;
- независимый от устройств обмен потоками данных в реальном времени;
- анализа поведения приложений и обмен данными с ними;
- статическая конфигурация устройств ввода-вывода и приоритетов задач.

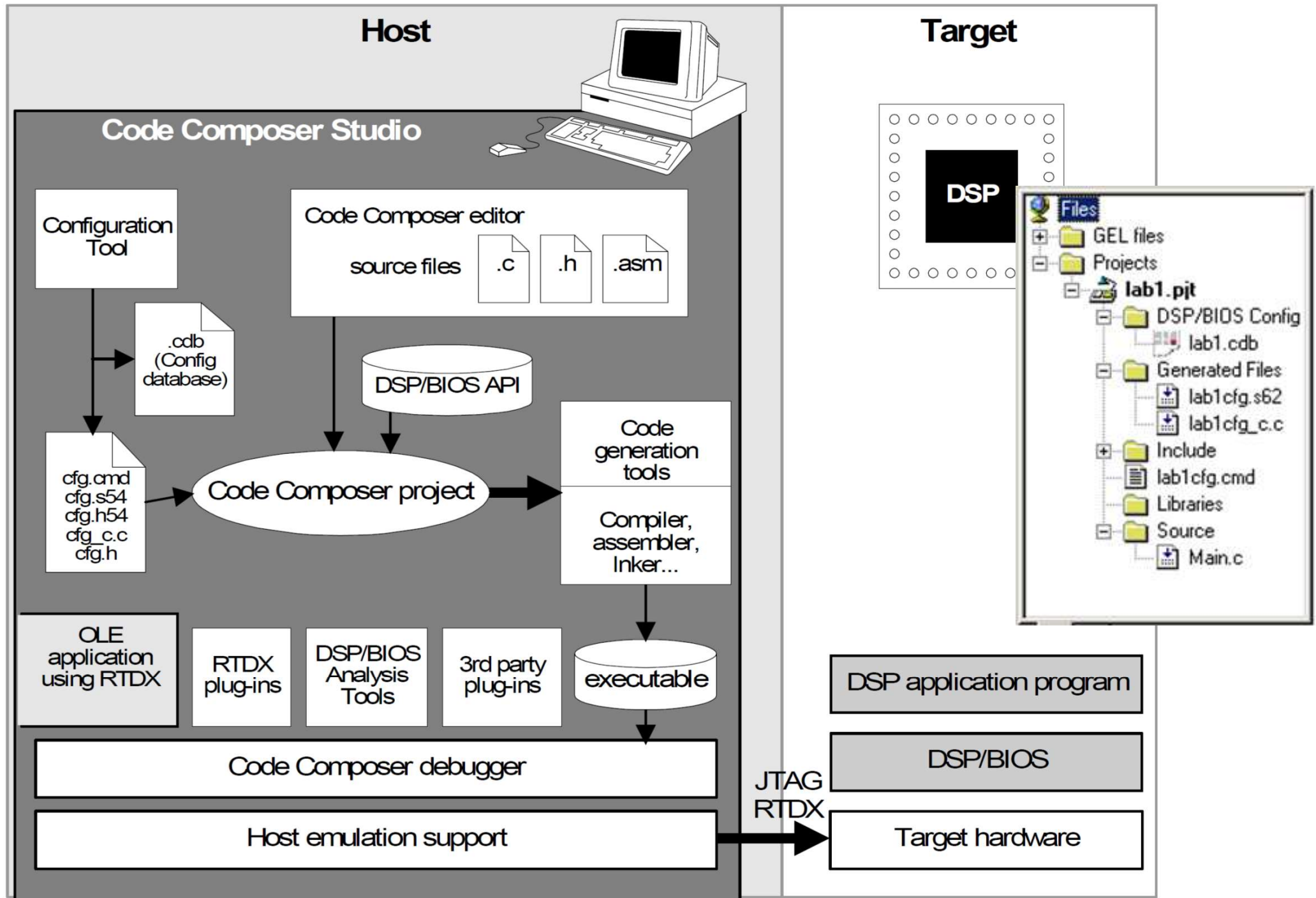
**Программные модули DSP/BIOS:**

- редактор конфигурирования;
- модуль анализа реального времени;
- модуль аппаратной абстракции;
- модуль ввода-вывода;
- модуль управления потоками;
- модуль синхронизации потоков.

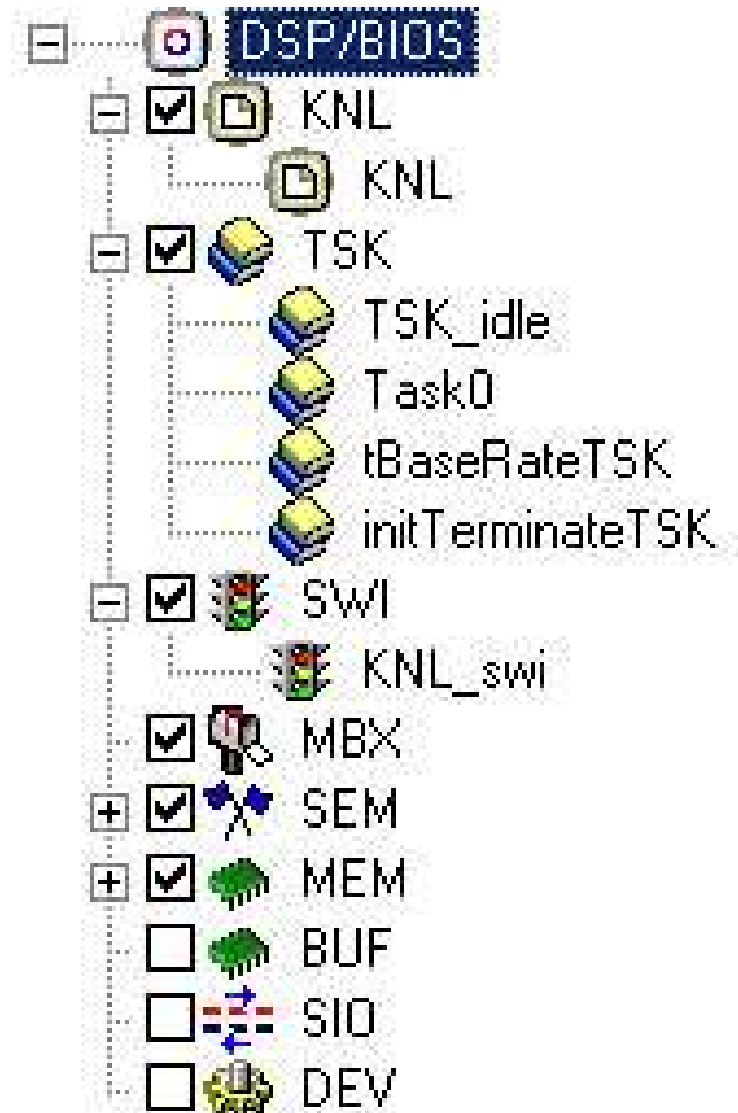
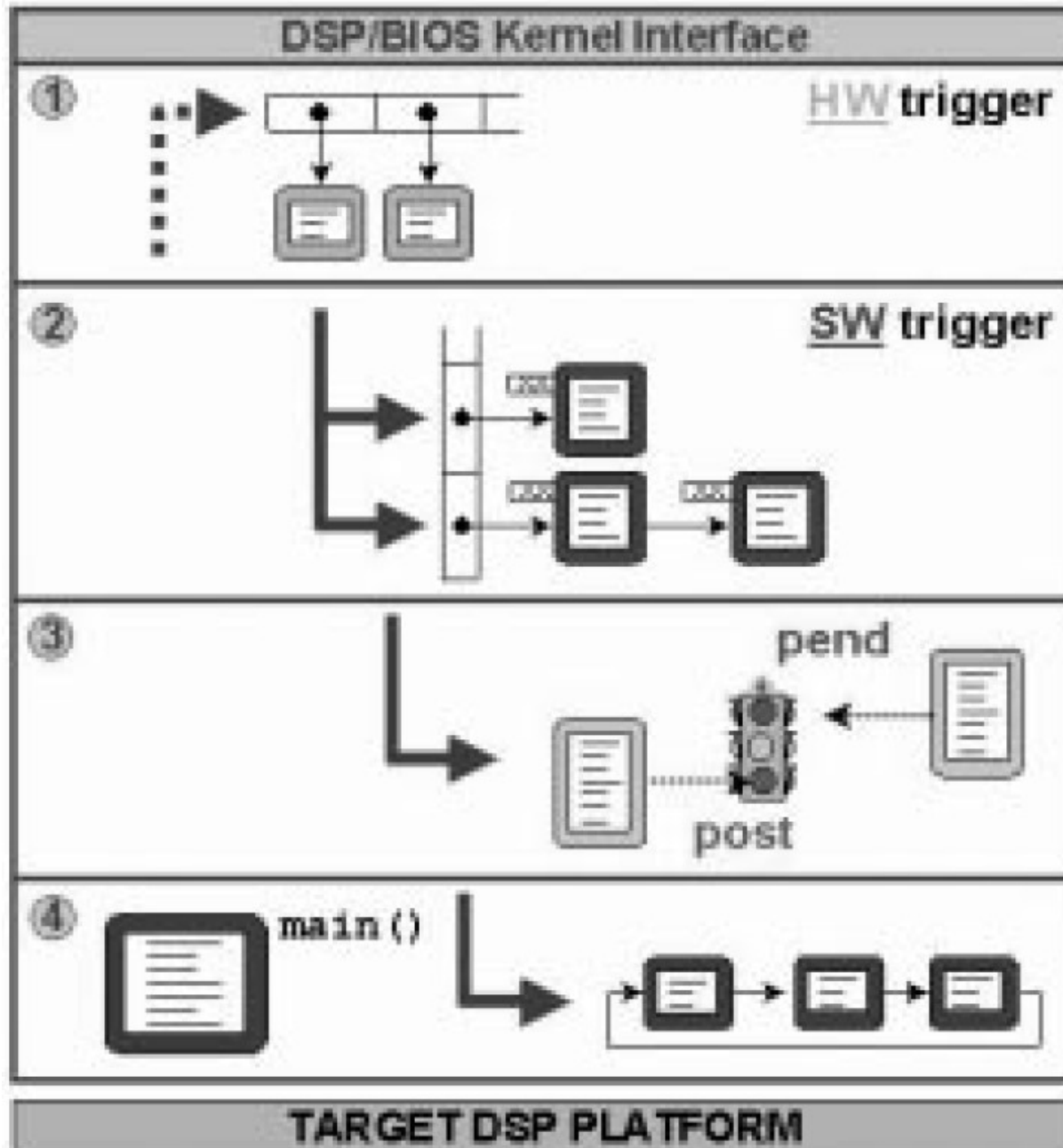




# Применение DSP/BIOS

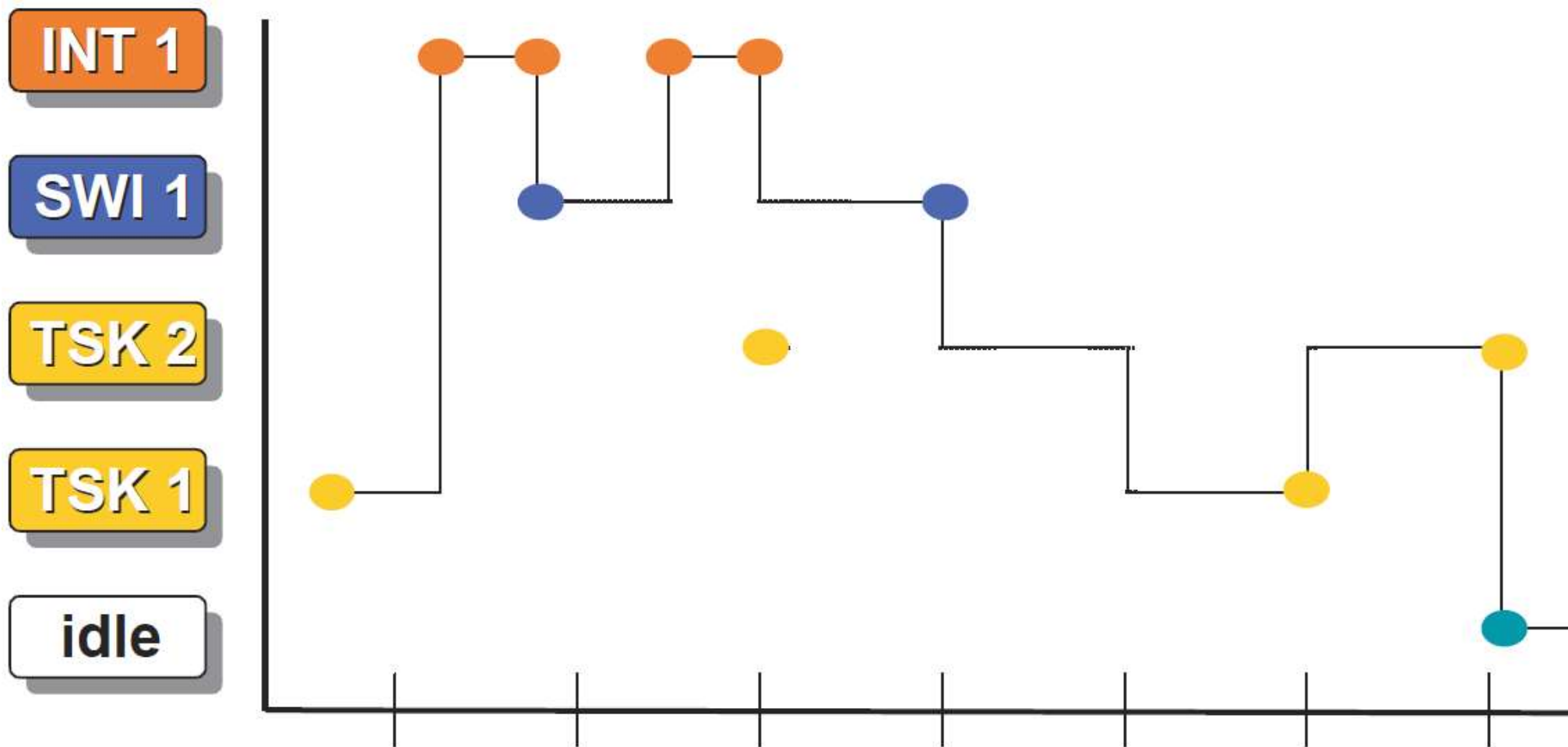


# Ядро DSP/BIOS



- ① **HARDWARE INTERRUPTS**
- ② **SOFTWARE INTERRUPTS**
- ③ **SYNCHRONIZED TASKS**
- ④ **BACKGROUND FUNCTIONS**

# Разделение времени





# Компоненты DSP/BIOS

## IA - Instrumentation/Real-Time Analysis

LOG	Message Log manger
STS	Statistics accumulator manager
TRC	Trace manager
RTDX	Real-Time Data Exchange manager

## TT - Thread Types

HWI	Hardware interrupt manager
SWI	Software interrupt manager
TSK	Multitasking manager
IDL	Idle function & processing loop manager

## PF - Clock and Periodic Functions

CLK	System clock manager
PRD	Periodic function manger

## CS - Comm/Synch threads

SEM	Semaphores manager
MBX	Mailboxes manager
LCK	Resource lock manager

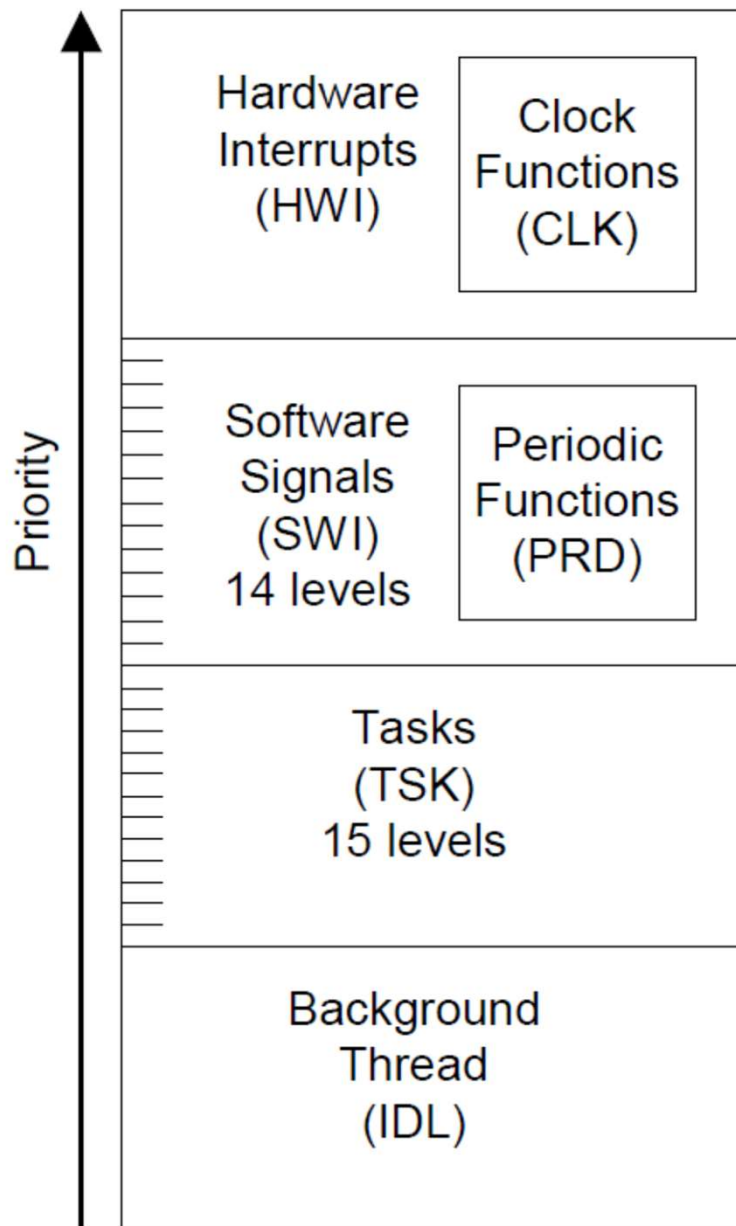
## IO - Input/Output

PIP	Data pipe manager
HST	Host input/output manager
SIO	Stream I/O manager
DEV	Device driver interface

## LP - Memory and Low-level Primitives

MEM	Memory manager
SYS	System services manager
QUE	Queue manager
ATM	Atomic functions
GBL	Global setting manager

# Thread Types (TT)

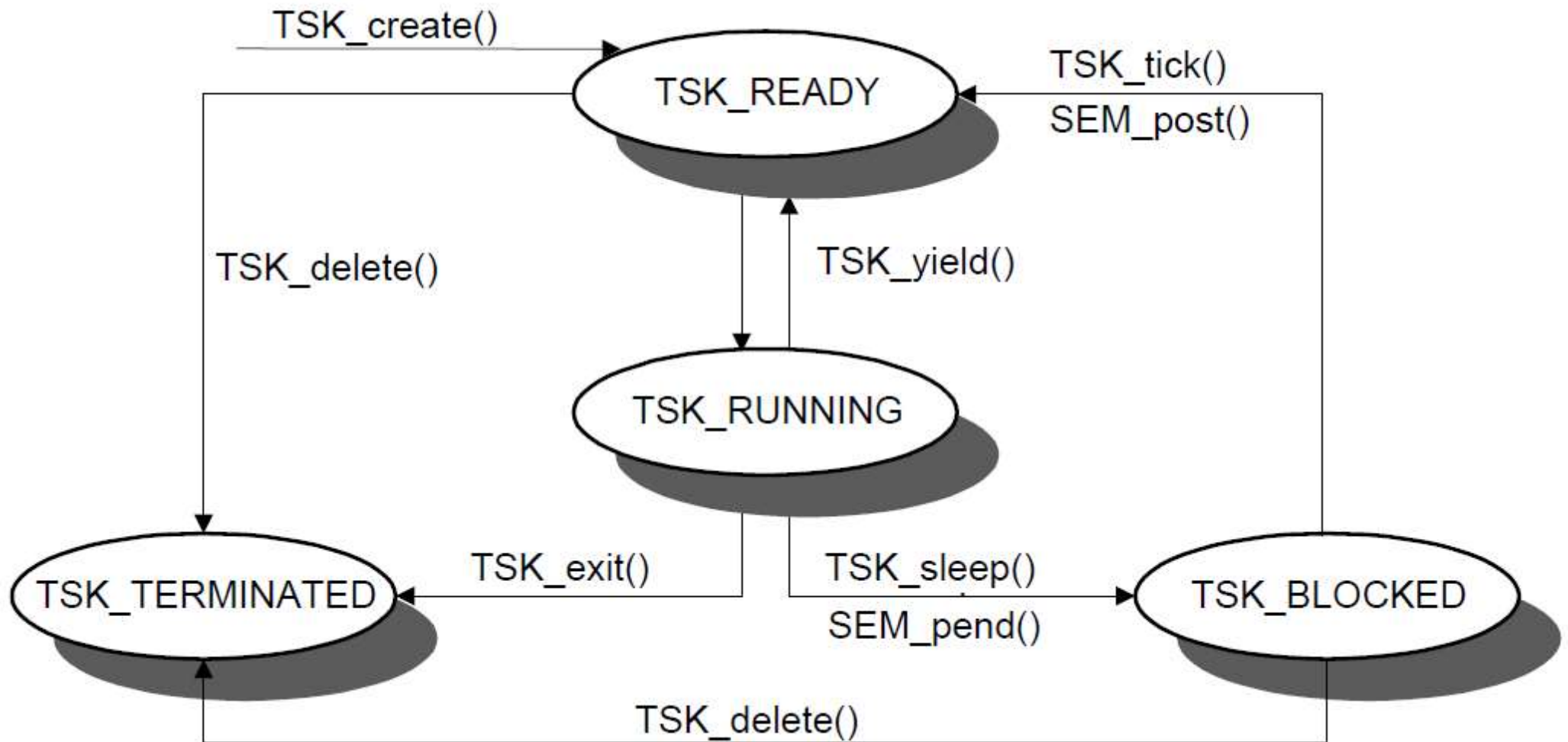


## Типы потоков (нитей):

- 1) аппаратные прерывания (HWI);
- 2) программные прерывания (SWI);
- 3) потоки задач (TSK)
- 4) фоновые потоки (IDL).

TSK_checkstacks	TSK_isTSK
TSK_create	TSK_itick
TSK_delete	TSK_self
TSK_deltatime	TSK_setenv
TSK_disable	TSK_seterr
TSK_enable	TSK_setpri
TSK_exit	TSK_settime
TSK_getenv	TSK_sleep
TSK_geterr	TSK_stat
TSK_getname	TSK_tick
TSK_getpri	TSK_time
TSK_getsts	TSK_yield

# ТТ – Состояния потоков



**Выполнение** (поток выполняться и занимает процессор).

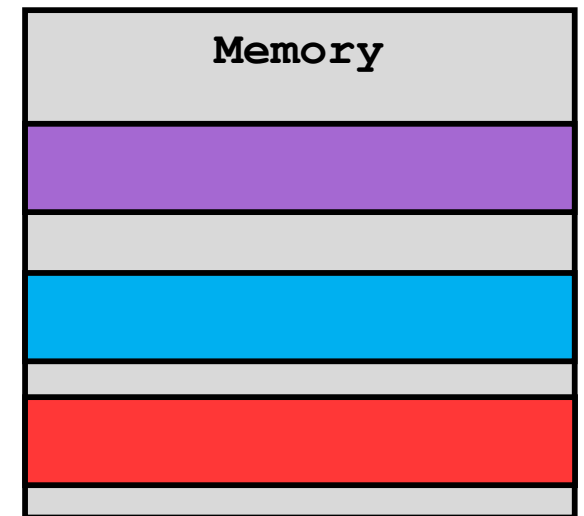
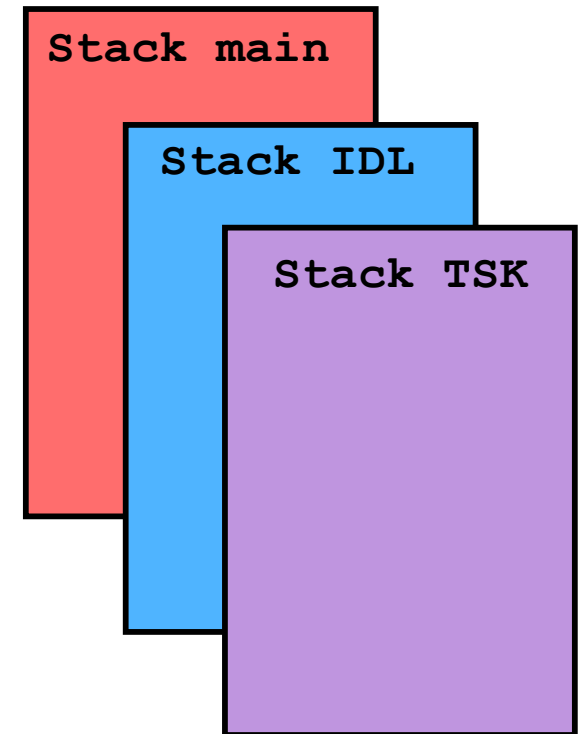
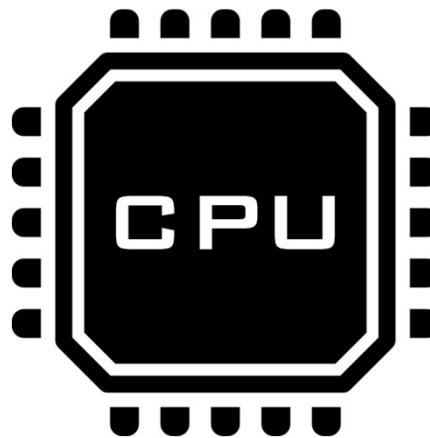
**Готовность** (поток находится в очереди на исполнение).

**Блокировка** (поток ожидает системного события).

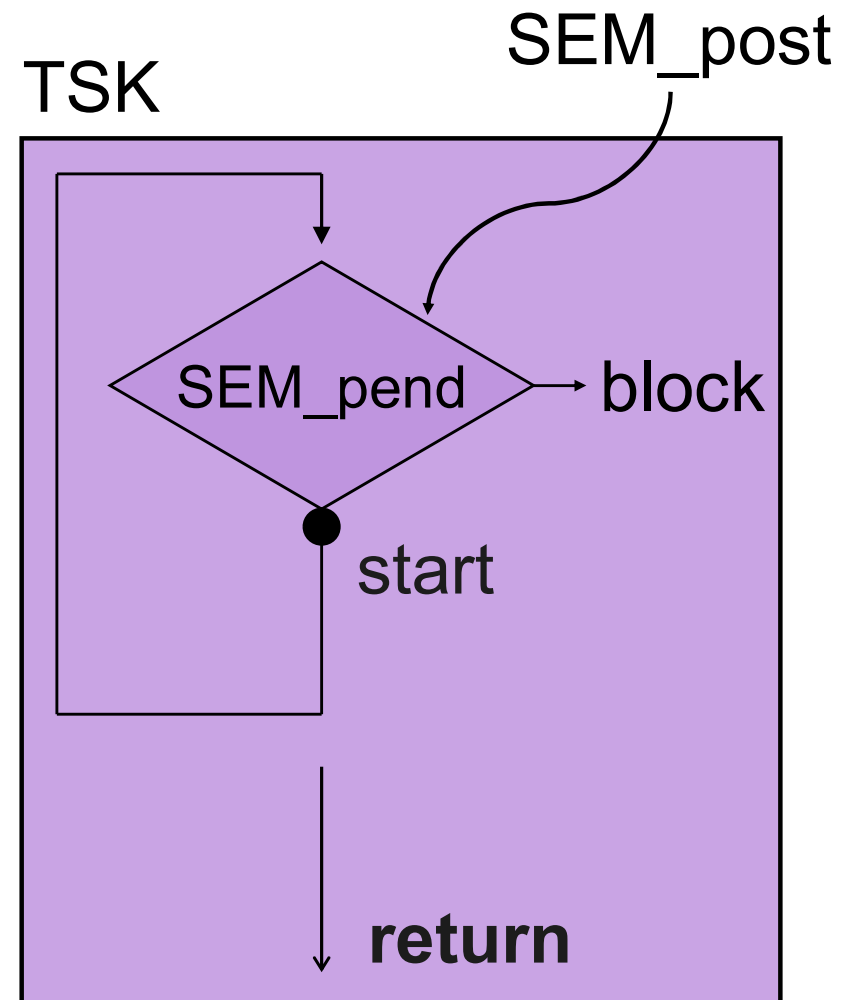
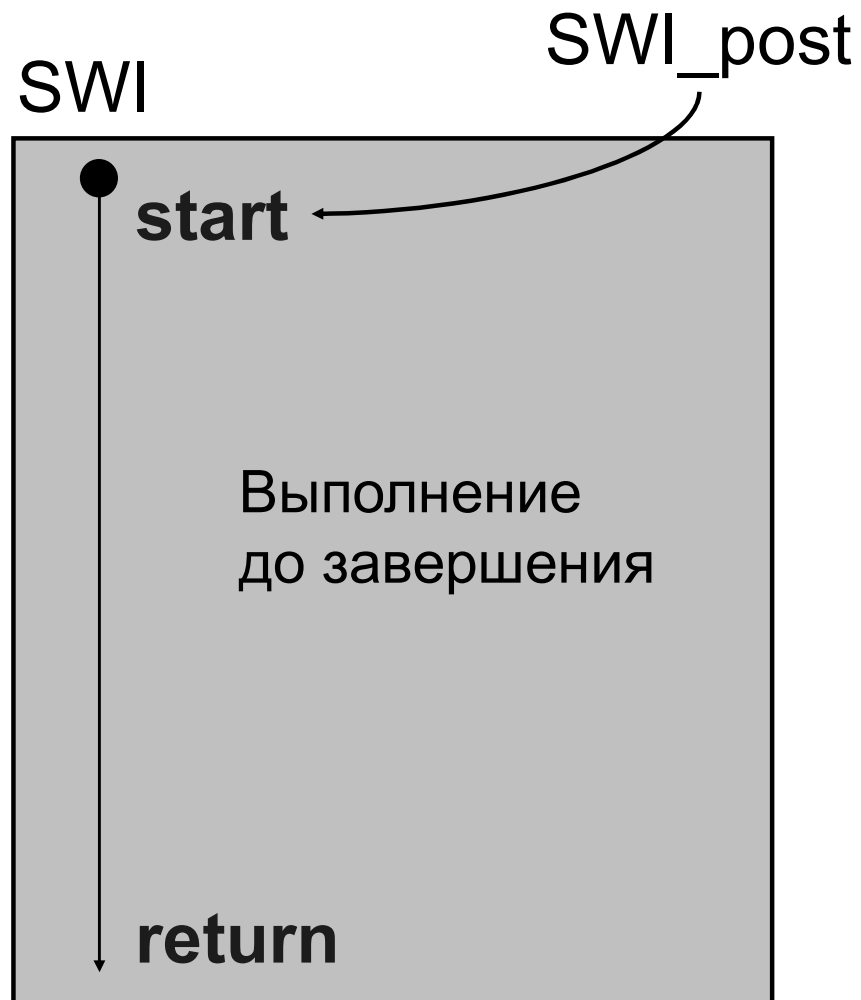
**Завершение** (поток завершен и больше не выполняется).

# ТТ – Ресурсы потоков

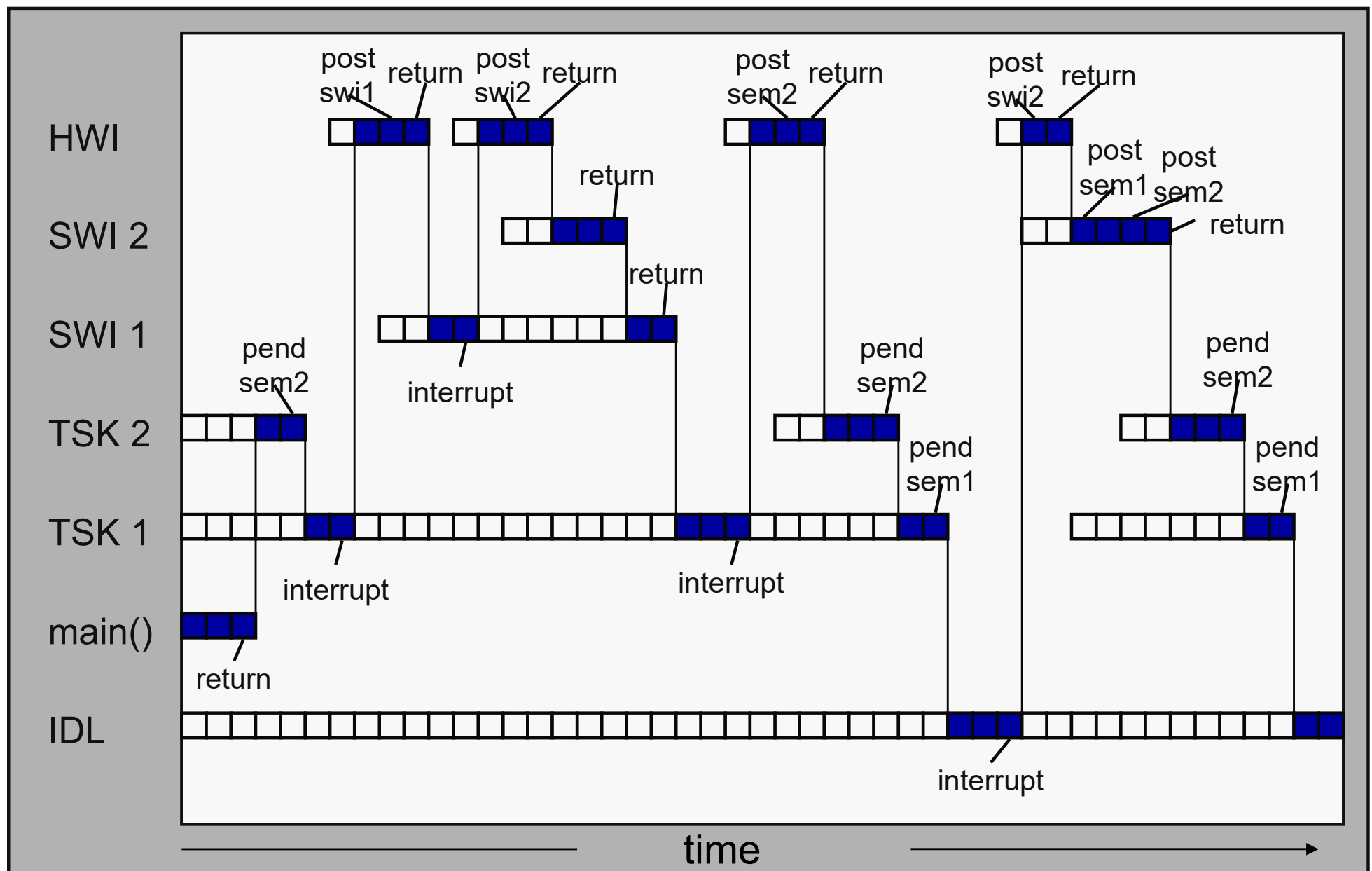
```
int main(int num, ...)
{
    int x, y[1024];
}
void IDL_THREAD()
{
    int x, y[1024];
}
void TSK_THREAD()
{
    int x, y[1024];
}
void SWI_THREAD()
{
    int x, y[1024];
}
void HWI_THREAD()
{
    int x, y[1024];
}
```



# ТТ – Переключение потоков



# ТТ – Выполнение потоков

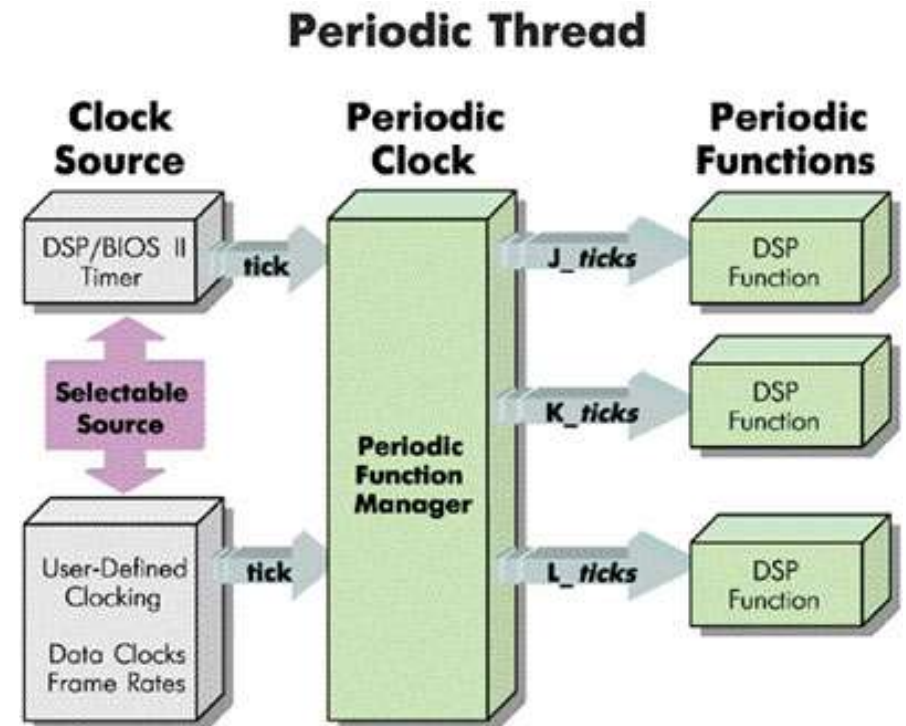
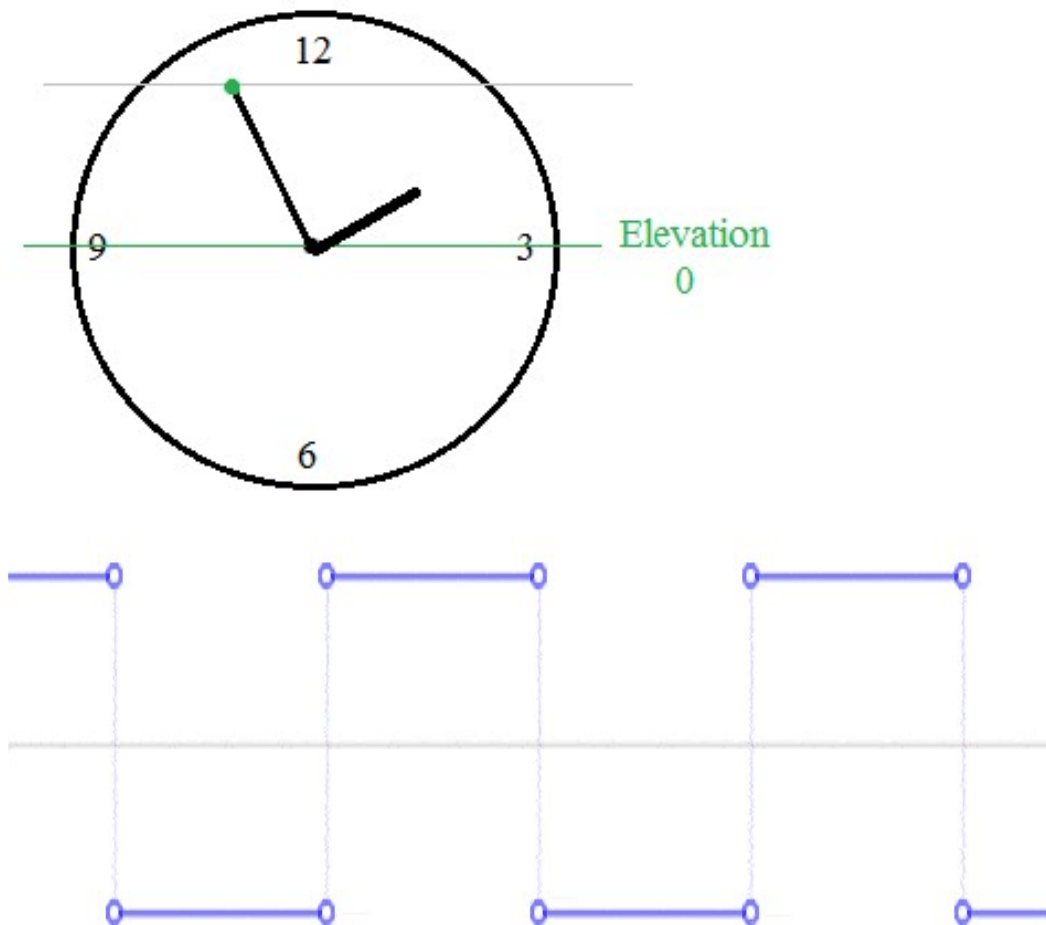




# Clock & Periodic Functions (PDF)

## Функции времени и периодов:

- 1) PRD\_getticks – получить номер текущего периода (тика);
- 2) PRD\_start – задать период исполнения периодической функции;
- 3) PRD\_stop – сбросить период исполнения периодической функции;
- 4) PRD\_tick – разрешение выполнения периодической функции.



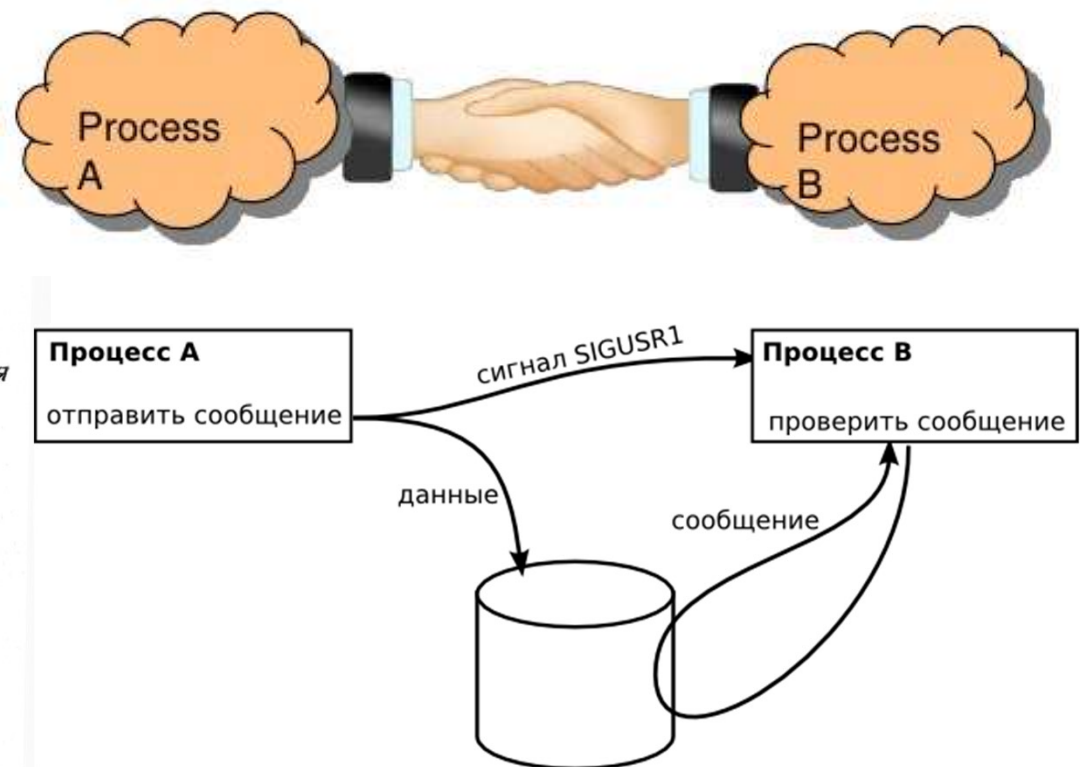
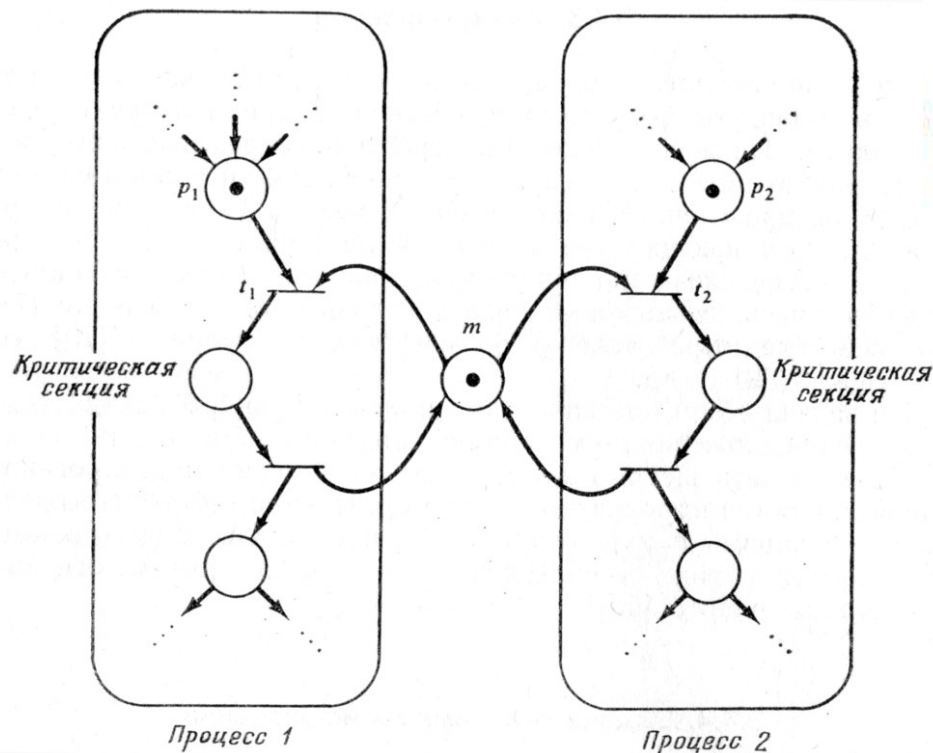
# Comm/Synch threads (CS)

## Примитивы синхронизации:

- 1) Semaphore (семафор) – блокировка потока пока семафор равен нулю.
- 2) Mailbox (почтовый ящик) – блокировка потока пока не отправлены данные;
- 3) Queue (очередь, массив почтовых ящиков) – блокировка пока нет данных.
- 4) Lock (блокировка).

## Примитивы коммуникации:

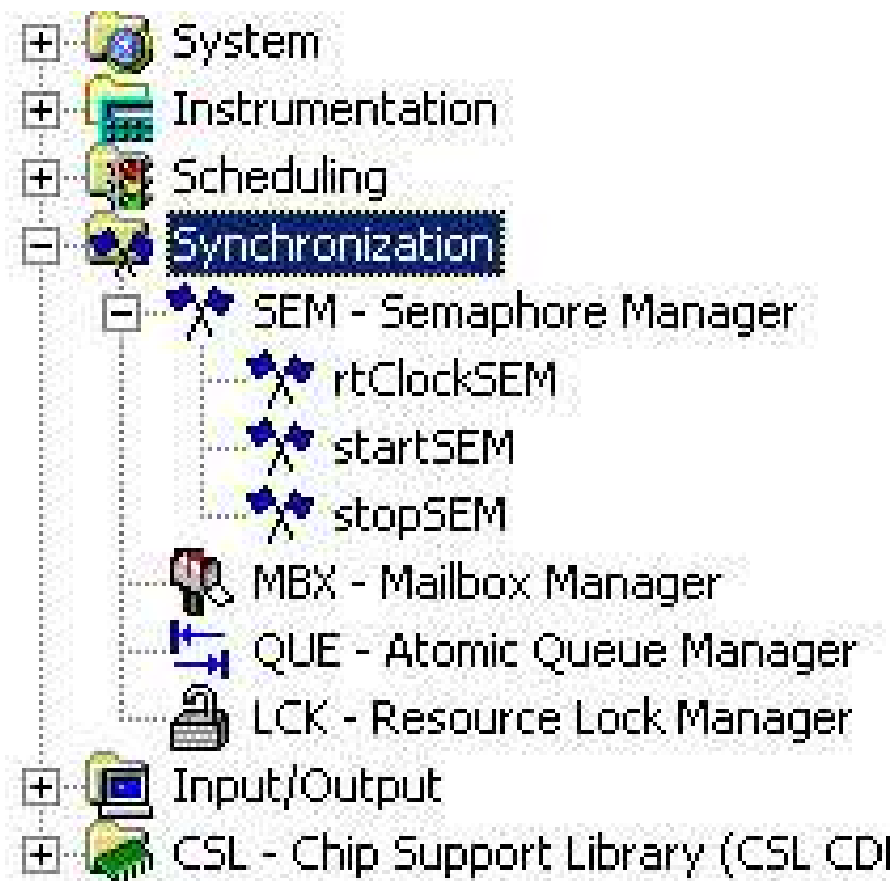
- 1) Pipe (канал) – обмен данными между потоками путем чтения-записи;
- 2) Stream (поток) – обмен между потоком и устройством.



# CS – Семафоры

## Функции семафоров:

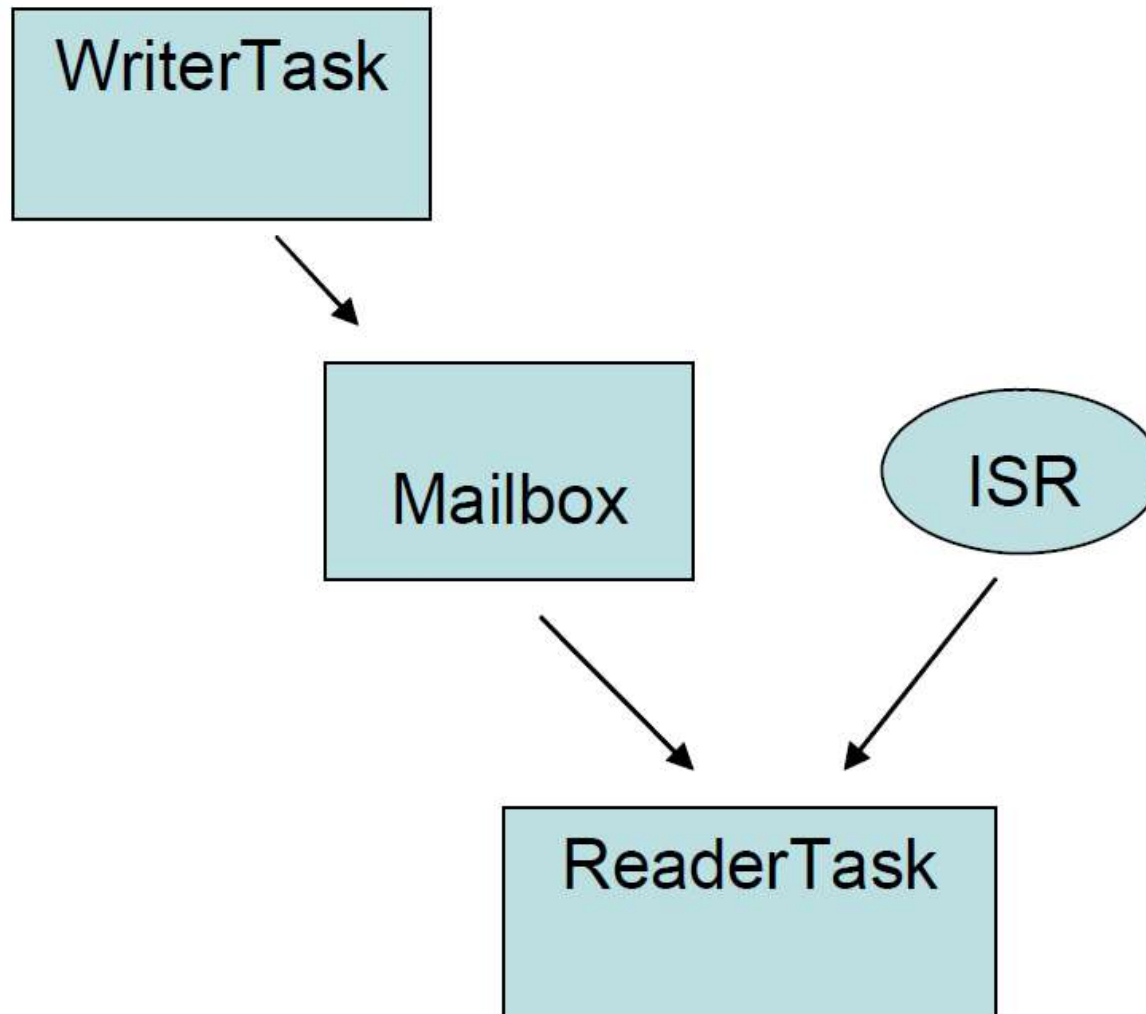
- 1) SEM\_create( ) – создание семафора;
- 2) SEM\_delete( ) – удаление семафора;
- 3) SEM\_pend( ) – ожидание на семафоре;
- 4) SEM\_post( ) – сигнал семафору;
- 5) SEM\_ipost( ) – сигнал семафору (для процедур прерывания);
- 6) SEM\_reset( ) – сброс семафора;
- 7) SEM\_new( ) – инициализация семафора;
- 8) SEM\_count( ) – счетчик семафора.



# CS – Почтовые ящики

## Функции почтовых ящиков:

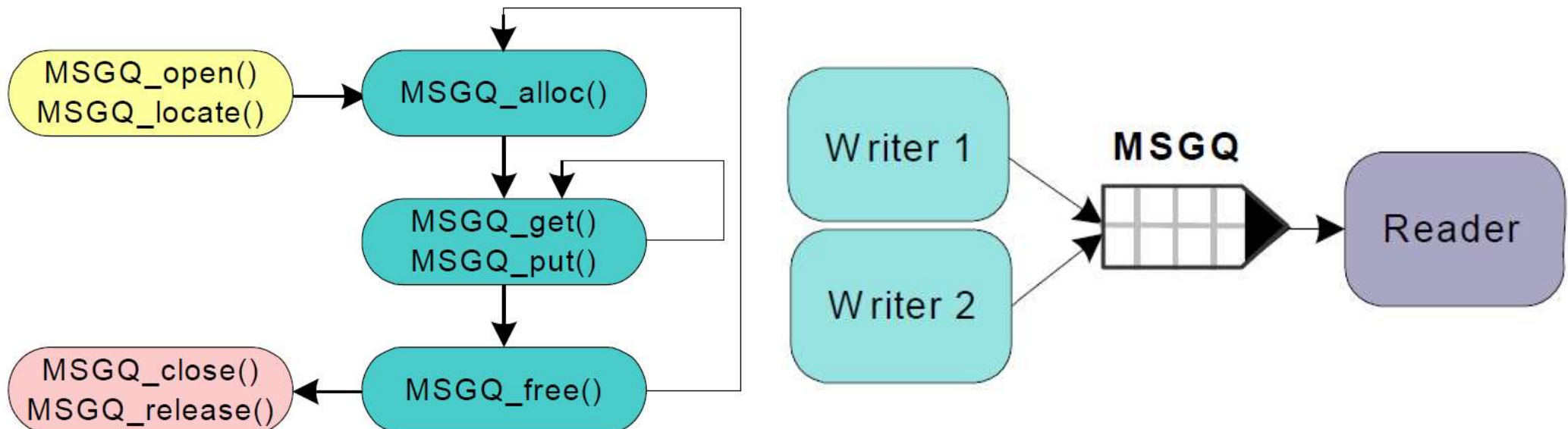
- 1) MBX\_create – создание почтового ящика;
- 2) MBX\_delete – удаление почтового ящика;
- 3) MBX\_pend – ожидание сообщения в почтовом ящике;
- 4) MBX\_post – передача сообщения в почтовой ящик.



# CS – Очереди

## Функции почтовых ящиков:

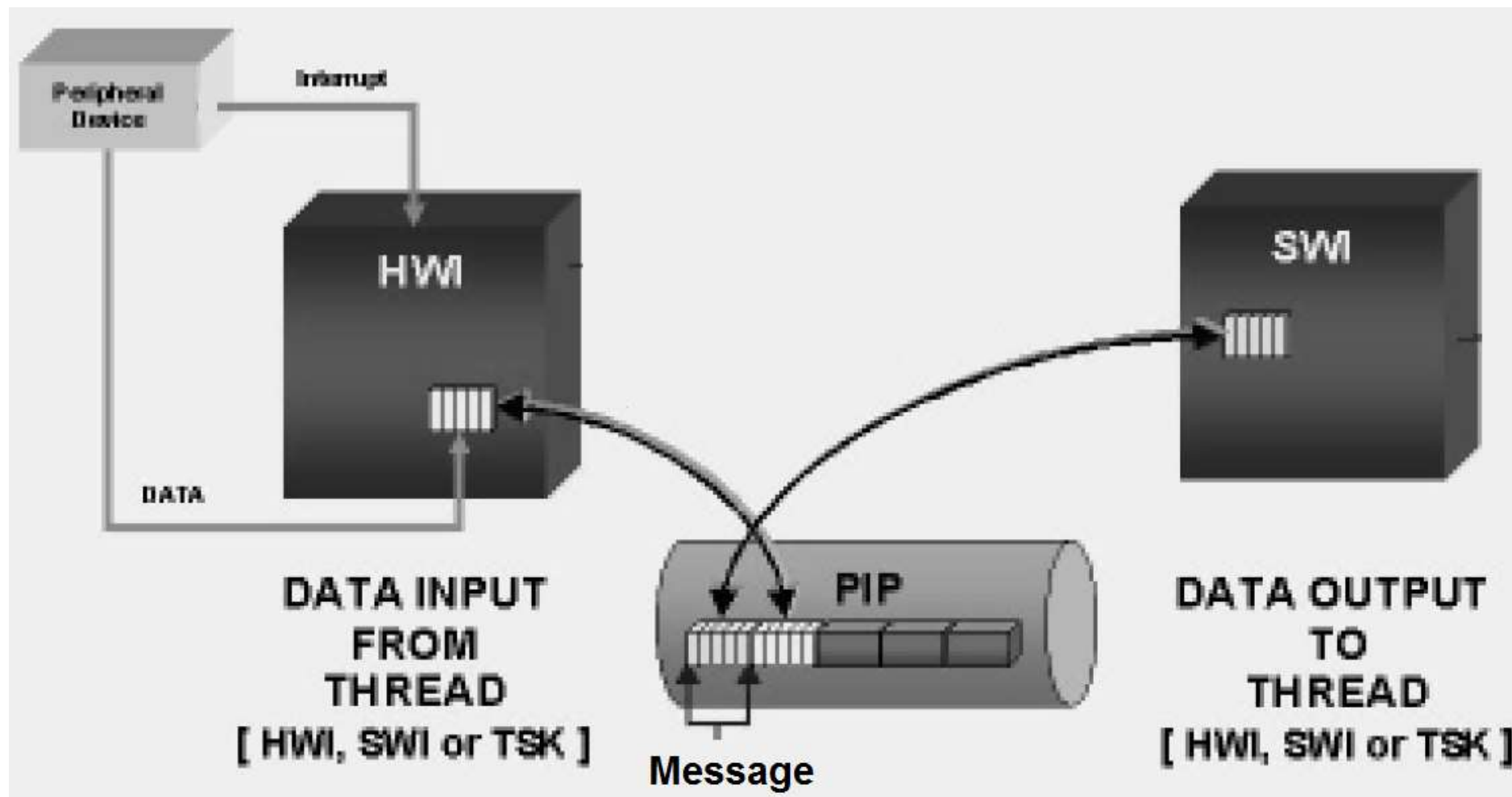
- 1) MSGQ\_open – создание очереди (читатель);
- 2) MSGQ\_close – удаление очереди (читатель);
- 3) MSGQ\_locate – подключение к очереди (писатель);
- 4) MSGQ\_release – отключение от очереди (писатель).
- 5) MSGQ\_alloc – создать сообщение (писатель);
- 6) MSGQ\_free – удалить сообщение (читатель).
- 7) MSGQ\_get – получение сообщения (читатель);
- 8) MSGQ\_put – передача сообщения (писатель).



# CS – Каналы

## Функции каналов:

- 1) PIP\_alloc – создание канал;
- 2) PIP\_free – освобождение канала;
- 3) PIP\_get – передать данные;
- 4) PIP\_put – получить данные.

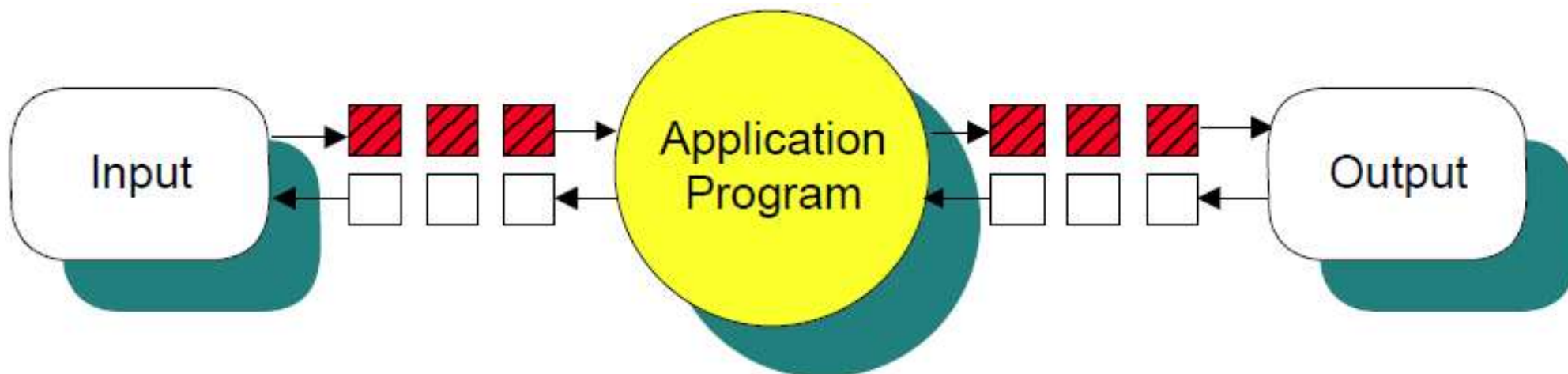




# CS – Потоки

## Функции потока:

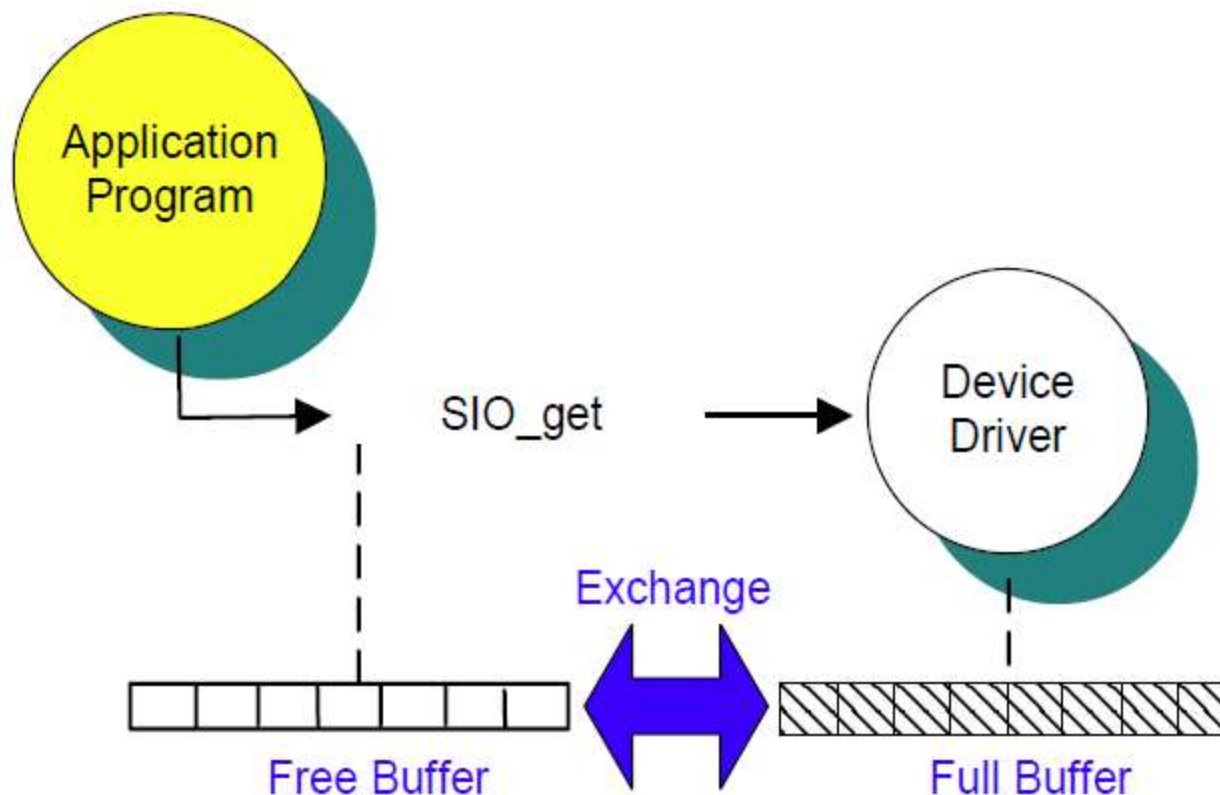
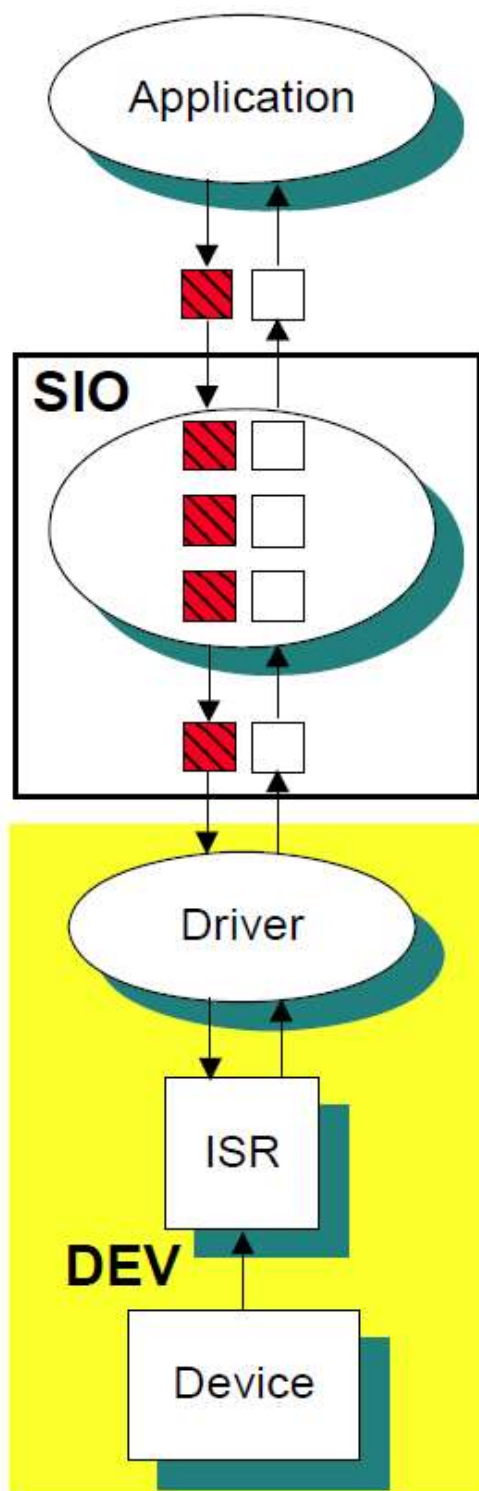
- 1) SIO\_create – создание потока;
- 2) SIO\_delete – уничтожение потока;
- 3) SIO\_get – обмен буфера получателем из потока;
- 4) SIO\_put – обмен буфера заполнителем потока;
- 5) SIO\_ctrl – управление потоком;
- 6) SIO\_idle – приостановка потока;
- 7) SIO\_flush – освобождение потока;
- 8) SIO\_select – ожидание буфера из потока;
- 9) SIO\_issue – занесение буфера в потока;
- 10) SIO\_reclaim – получение буфера из потока;
- 11) SIO\_staticbuf – передача буфера в поток и ожидание его обработки.



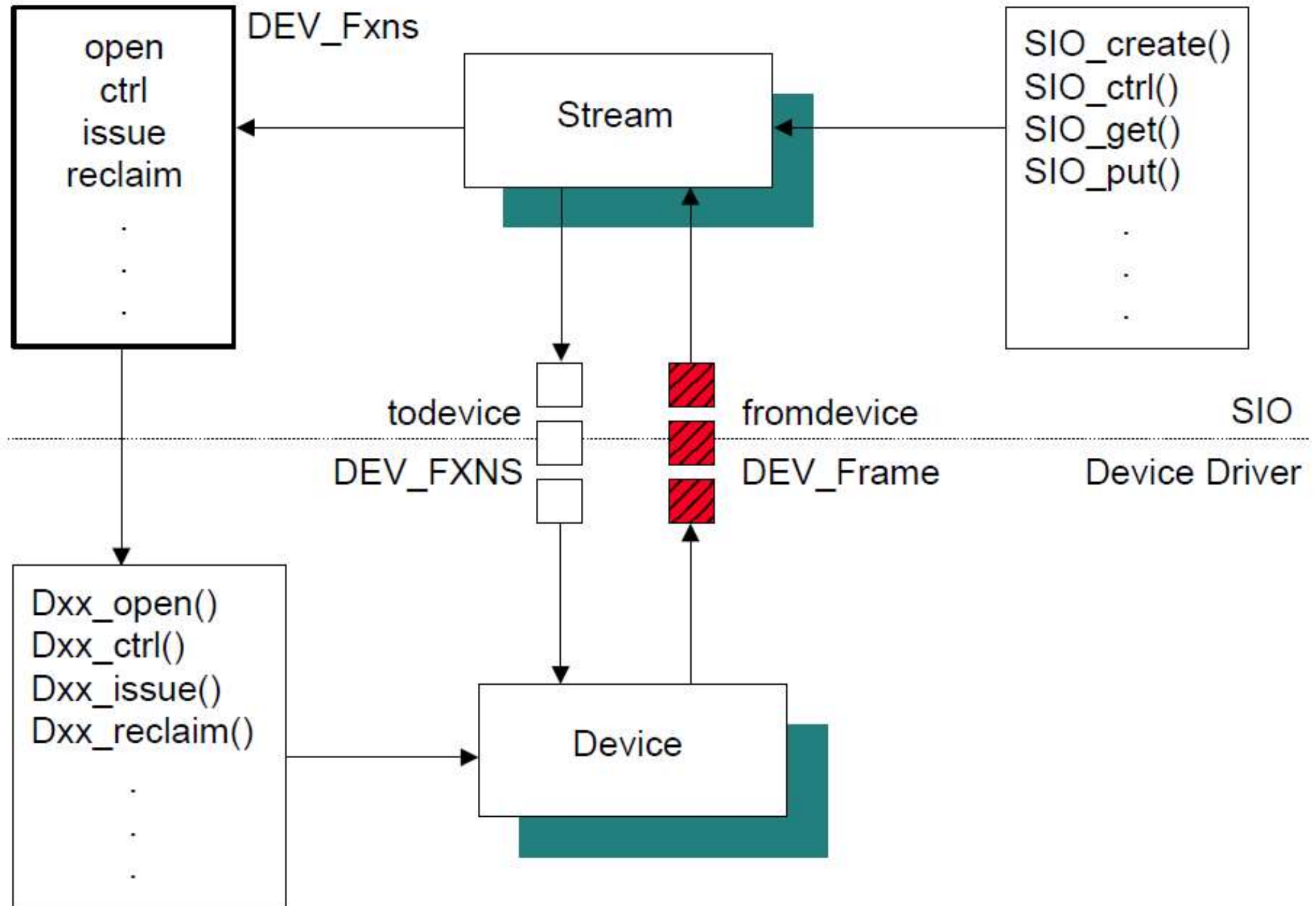
# CS – Устройства

## Функции устройств:

- 1) Dxx\_open – открыть устройство;
- 2) Dxx\_close – закрыть устройство;
- 3) Dxx\_issue – передать буфер устройству;
- 4) Dxx\_reclaim – получить буфер от устройства;
- 5) Dxx\_ctrl – управление устройством;
- 6) Dxx\_idle – приостановка устройства;
- 7) Dxx\_ready – получение состояния устройства.



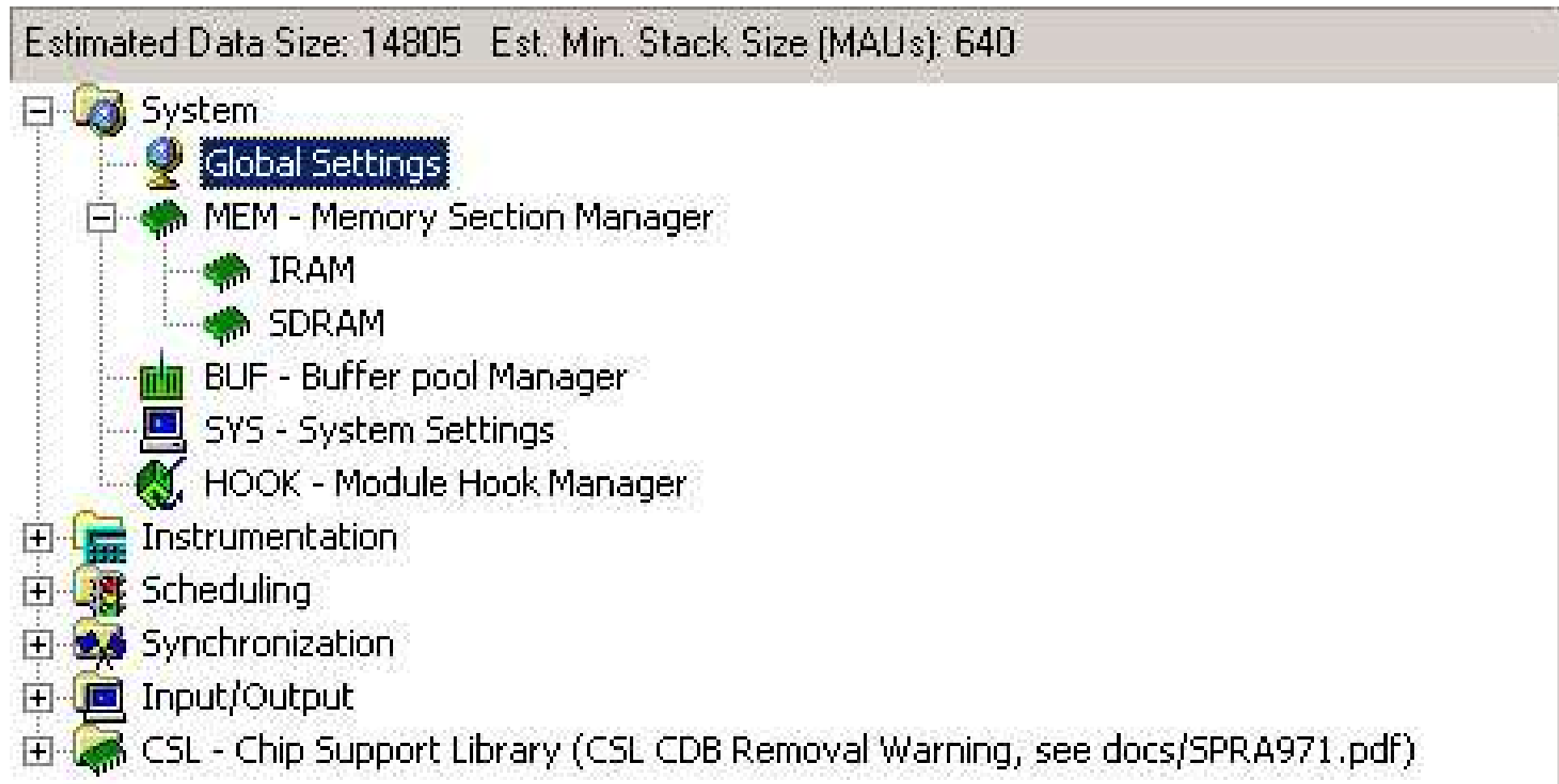
# CS – Поток и устройства



# Memory and Low-level Primitives

## Низкоуровневые функции:

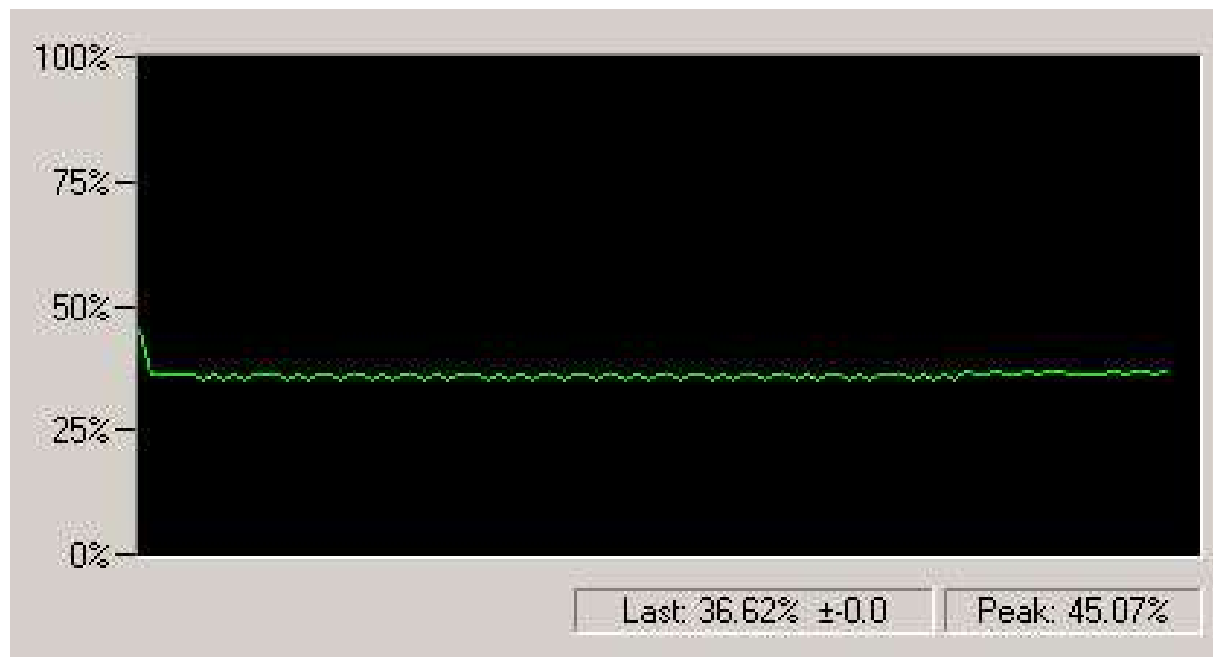
- 1) MEM\_alloc – выделение блока памяти;
- 2) MEM\_free – освобождение блока памяти;
- 3) MEM\_stat – состояние памяти;
- 4) MEM\_post – передача сообщения в почтовой ящик.



# Instrumentation/Real-Time Analysis

## Сервисы анализа:

- 1) Message Event Log (журнал сообщений) – сбор и предоставление данных о событиях, формируемых потоками.
- 2) Statistic Accumulator (сбор статистики) – сбор и предоставление динамических параметров (счетчики, времена выполнения, объемы ресурсов);
- 3) Trace (трассировка) – отслеживание поведения потоков в реальном времени;
- 3) Real-Time Data Exchange (каналы обмена данными с главным компьютером) – потоки данных для тестирования и анализа поведения потоков и ресурсов.

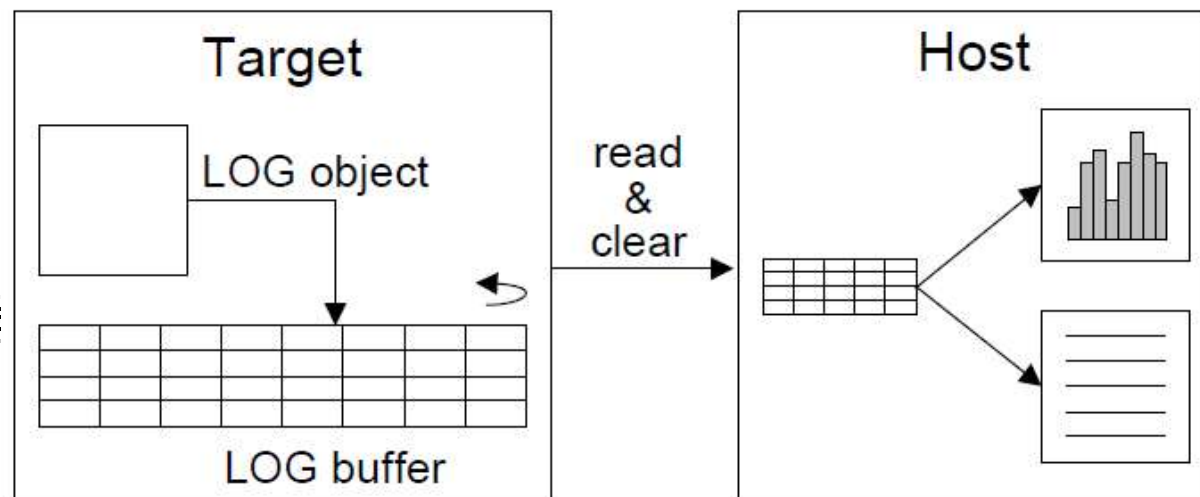
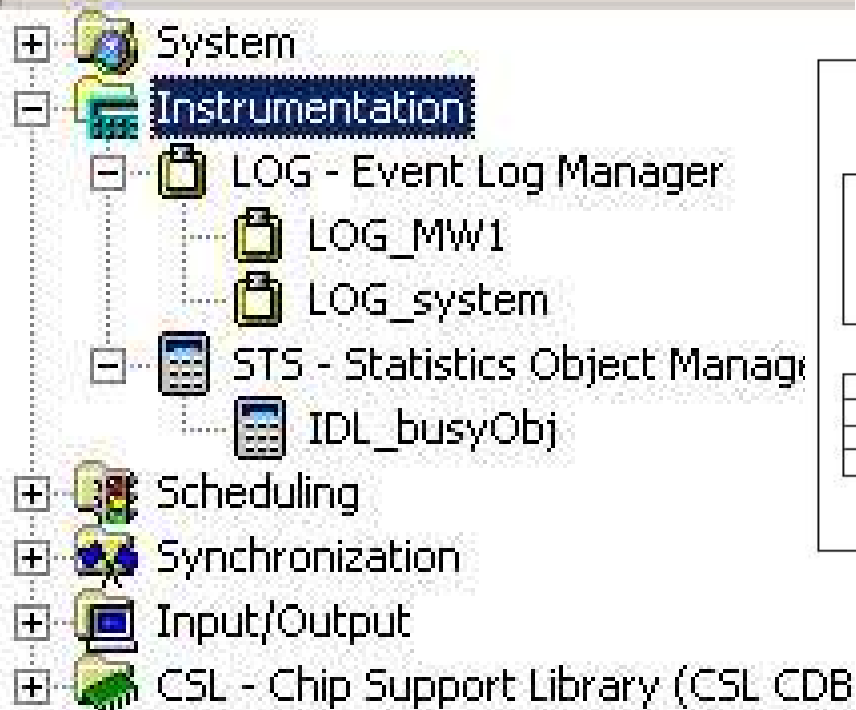


# LOG – Message Log manger

## Функции протоколирования:

- 1) LOG\_disable – Disable the system log;
- 2) LOG\_enable – Enable the system log;
- 3) LOG\_error – Write a user error event to the system log;
- 4) LOG\_event – Append unformatted message to a message log;
- 5) LOG\_message – Write a user message to the system log;
- 6) LOG\_printf – Append a formatted message to a message log;
- 7) LOG\_reset – Reset the system log.

Estimated Data Size: 14805 Est. Min. Stack





# STS – Statistics accumulator manager

## Функции статистики:

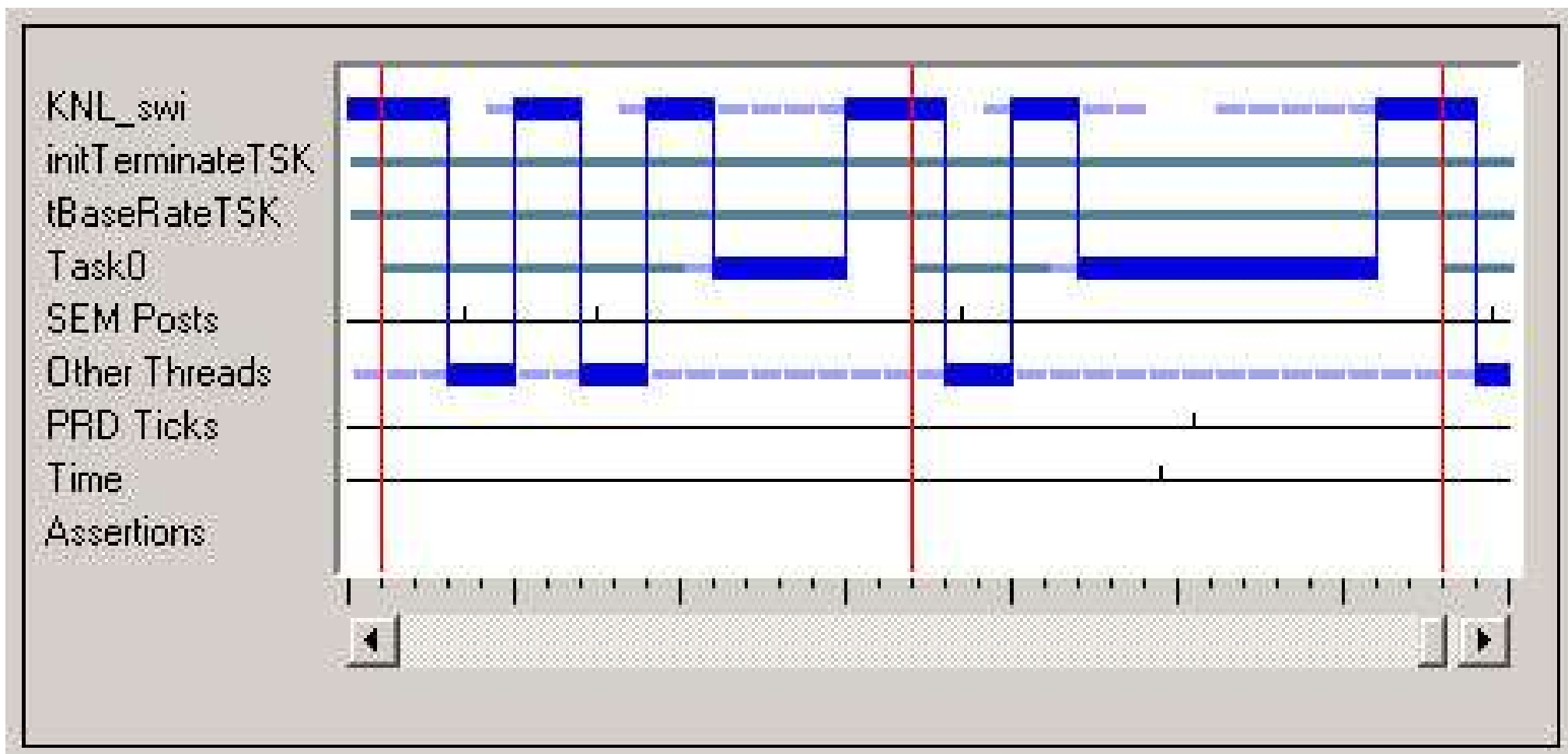
- 1) STS\_add( ) – обновление статистики (значение);
- 2) STS\_delta( ) – обновление статистики (изменение);
- 3) STS\_reset( ) – сброс значения статистики;
- 4) STS\_set( ) – сохранения статистики.

## Типы статистик:

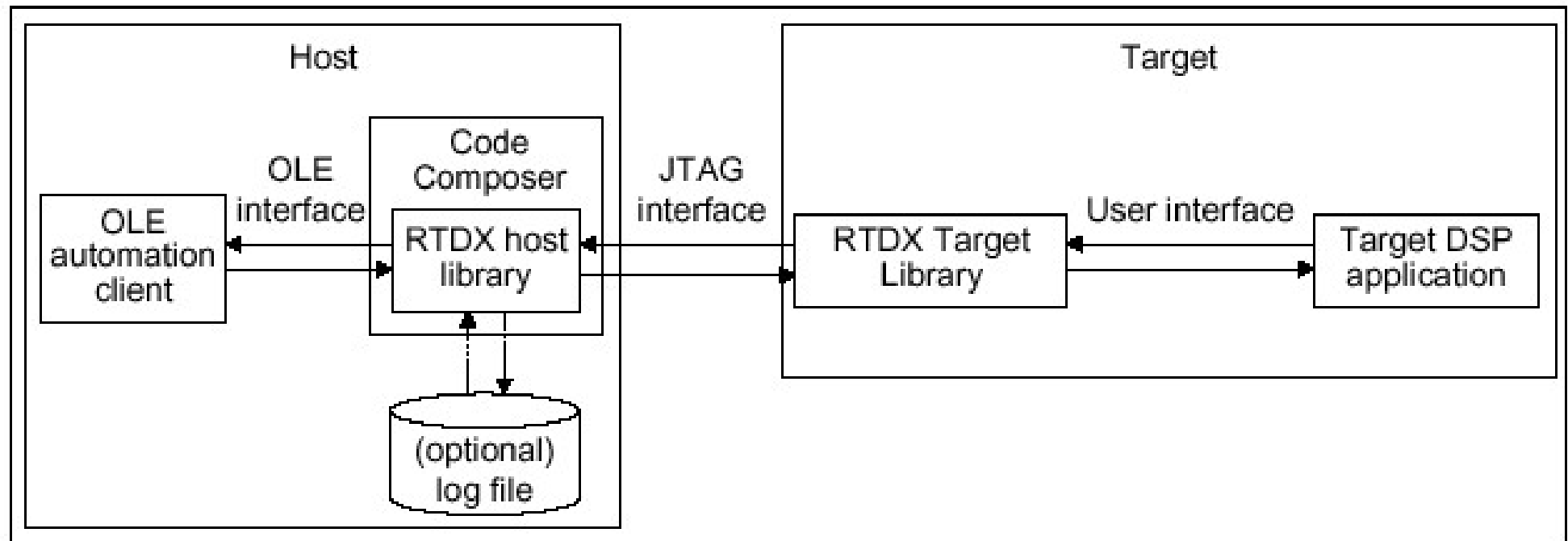
- 1) Count (счетчик) – число значений;
- 2) Total (всего) – сумма значений;
- 3) Average (среднее) – среднее значение;
- 4) Maximum (максимум) – максимальное значение.

STS	Count	Total	Max	Average
initTerminateTSK	0	0 inst	-2147483648 inst	0.00
tBaseRateTSK	0	0 inst	-2147483648 inst	0.00
Task0	0	0 inst	-2147483648 inst	0.00
TSK_idle	0	0 inst	-2147483648 inst	0.00
IDL_busyObj	416019	-4.59118e+007	-71	-110.36

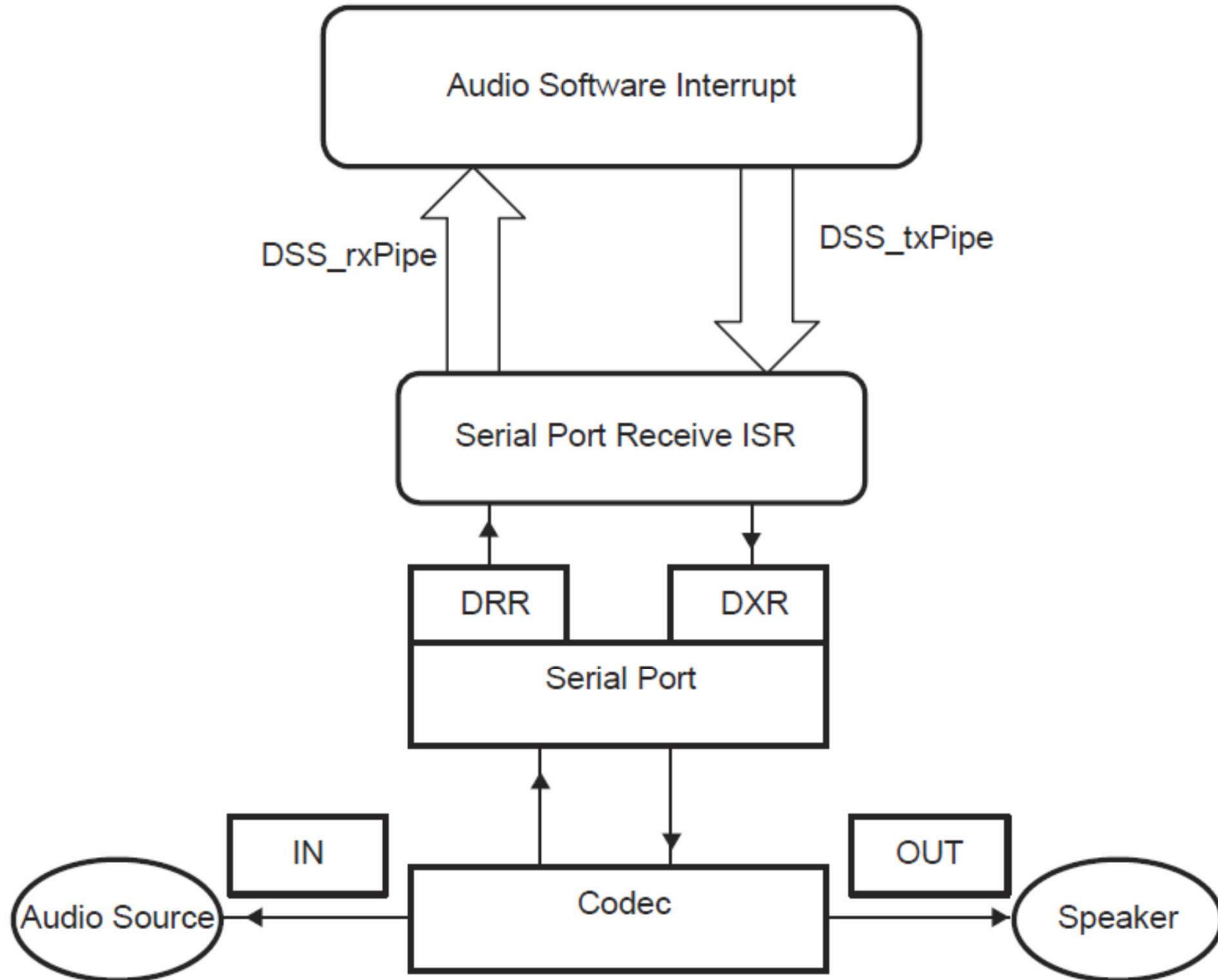
# TRC – Trace manager



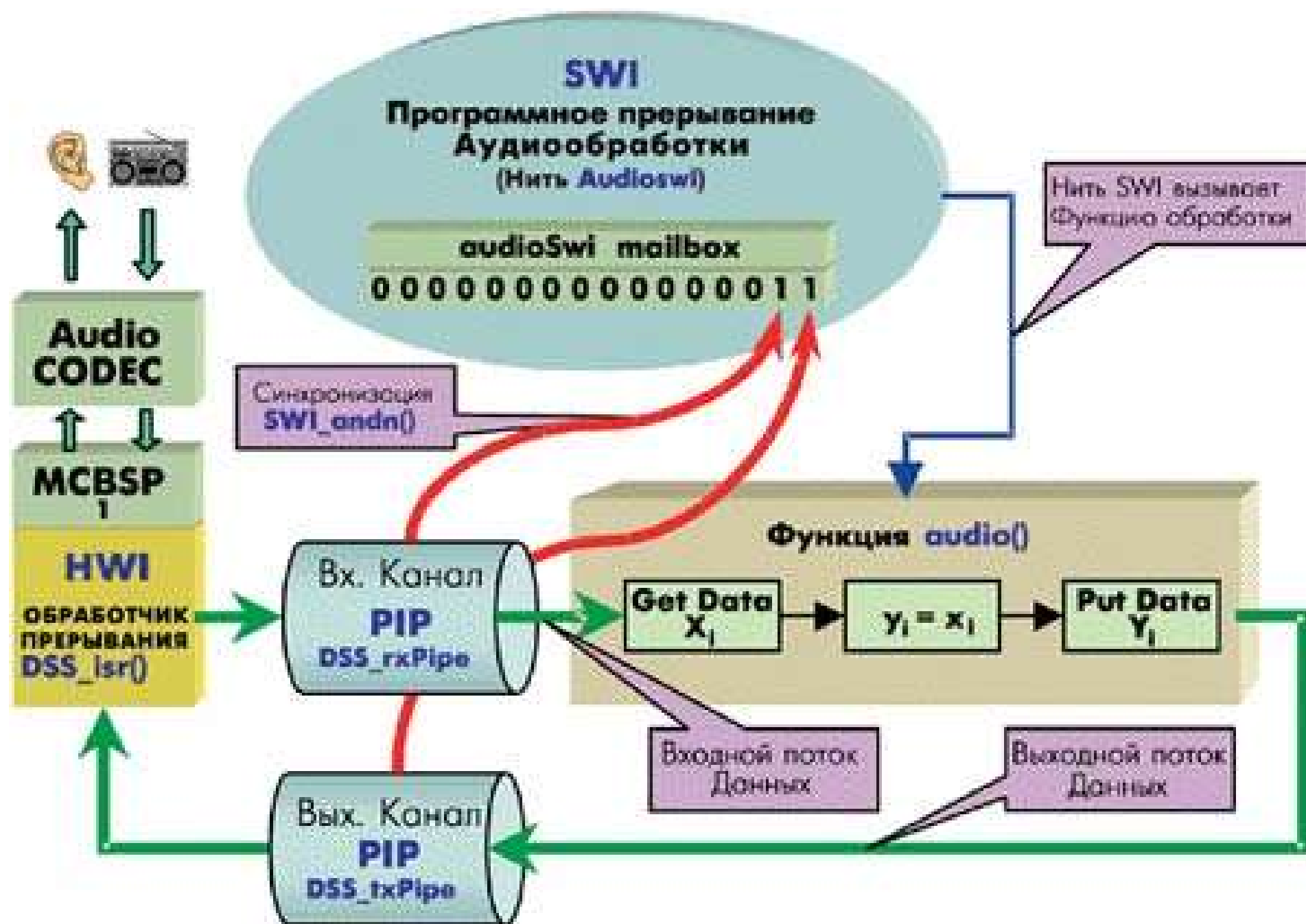
# RTDX – Real-Time Data Exchange manager



# Приложение

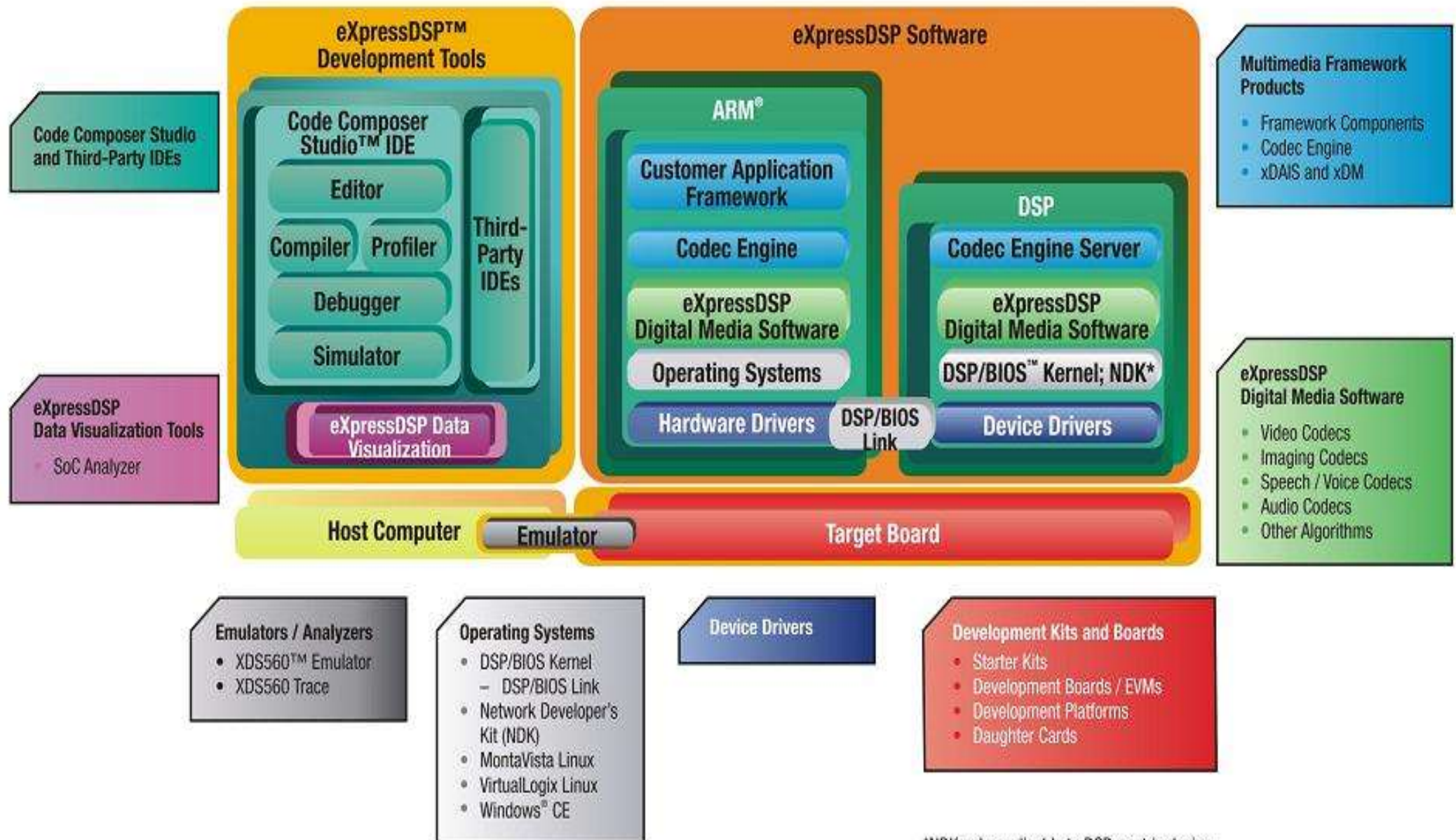


# Программа



# ARM и DSP

## SoC (ARM® + DSP)



\*NDK only applicable to DSP-centric design