

# Шина PCI Express

# Шина PCI Express

Шина PCI Express (проект Agarahoe) была разработана в 2002 году как универсальный периферийный интерфейс системного уровня. Первая общепринятая спецификация имеет версию 1.0a, она была принята комитетом PCI SIG в 2003 году. Позднее была принята спецификация 1.1, в 2007 году одобрена спецификация 2.0. Появление версии 3.0 ожидается в 2010 году.

При разработке PCI Express особое внимание было уделено совместимости с PCI на уровне механизма конфигурирования, программного доступа и поддержки со стороны ОС и драйверов. При этом требовалось сохранить или уменьшить стоимость реализации при значительном улучшении всех характеристик, прежде всего пропускной способности.

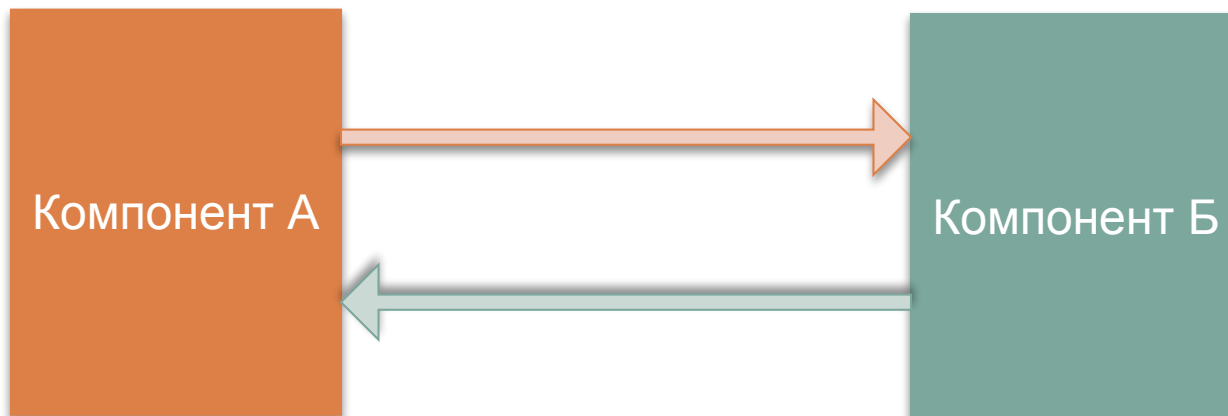
# PCI Express Link

Вместо шинного соединения PCI в PCI Express применена схема объединенных через коммутаторы двухточечных каналов связи между устройствами и портами.

Соединение (Link) – это две пары встречных симплексных каналов.

Каждый канал является низковольтной дифференциальной парой сигналов.

Скорость соединения (Signaling Rate) устанавливается в начале работы шины; определены две скорости – 2.5 Гбит/с и 5.0 Гбит/с (PCIe 2.0).



# PCI Express Lane

Соединение (Link) может включать одну или несколько линий (Lane), каждая из которых представляет собой пару дифференциальных сигналов – передающую (Transmitting) и принимающую (Receiving). В целях масштабирования соединение может агрегировать несколько линий.

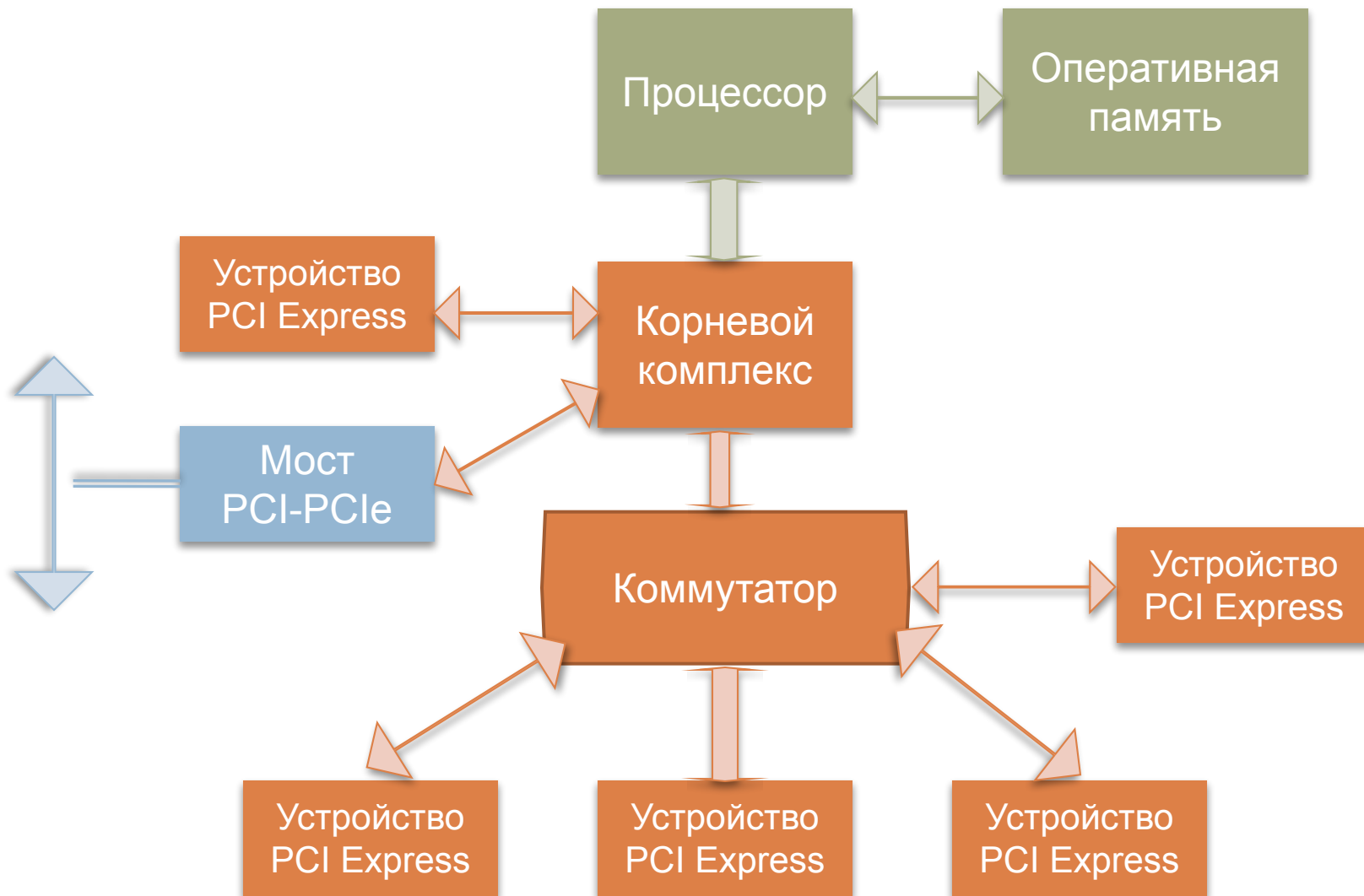
Спецификация предусматривает следующие конфигурации соединения:

x1, x2, x4, x8, x12, x16, x32.

Количество дифференциальных пар на прием и передачу должно быть одинаково, несимметричные соединения невозможны.

Данные по разным линиям передаются побайтно, общий поток делится на блоки, кратные количеству линий.

# Коммутационная фабрика PCI Express



## Корневой комплекс (RC)

Это аналог главного моста (Host Bridge) в шине PCI. Он отвечает за связь с процессором и системной памятью, а также за конфигурирование всей фабрики.

RC содержит несколько портов PCI Express (Root ports), которые могут (необязательно) взаимодействовать между собой посредством виртуального коммутатора. К каждому из портов RC может подключаться коммутатор (switch), мост для другой шины (напр., PCI) или конечное устройство (Endpoint).

RC отвечает за конфигурационные циклы, может выполнять циклы доступа к портам и пространству памяти. RC может запрашивать заблокированные (Locked) операции, но не может отвечать на запросы с блокировкой.

# Конечное устройство (Endpoint)

Каждое конечное устройство подключается к порту либо RC, либо коммутатора. Устройство выполняет транзакции от своего имени либо от имени подключенной к нему шины, устройства или контроллера другого интерфейса.

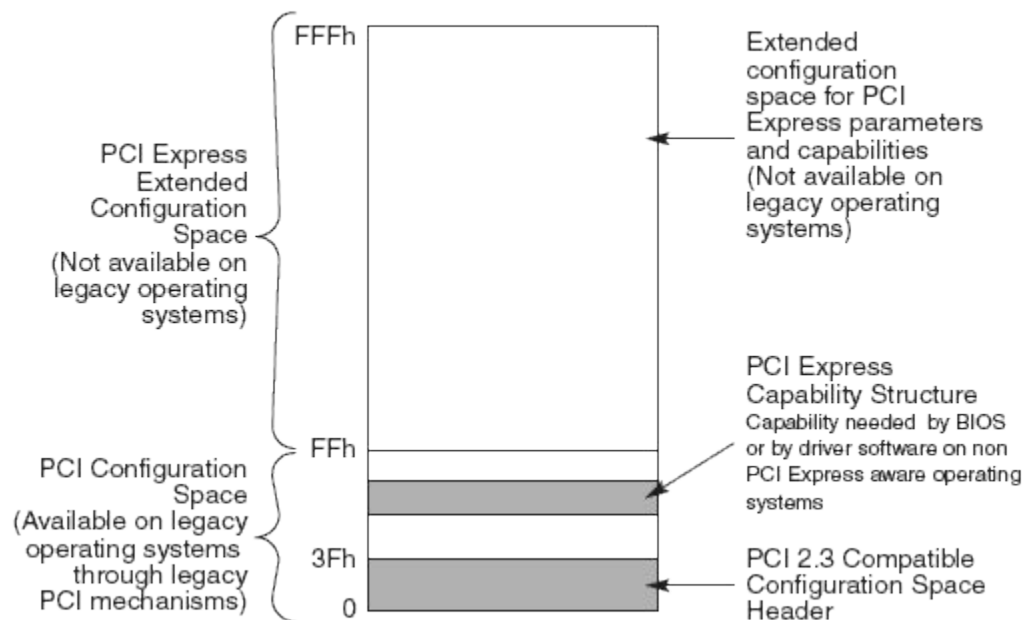
Устройства могут быть полноценными и устаревшего типа (Legacy).

Полноценное устройство:

- Не работает через порты – только через диапазон памяти
- Не работает с заблокированными запросами
- Поддерживает 64-битное адресное пространство по умолчанию
- Поддерживает механизм прерываний MSI, причем с 64-битным пространством
- Имеет расширенное пространство конфигурирования

# Расширенный механизм конфигурирования

Позаимствован у PCI-X 2.0. Стандартный способ доступа – через конфигурационный цикл – сохранен для совместимости. Полное конфигурационное пространство каждого устройства занимает 4 Кб.





## (продолжение)

Для упрощения доступа к конфиг. регистрам предусмотрен механизм их отображения на пространство памяти. По заданному базовому адресу находится пространство для всех возможных устройств в рамках системной шины.

Memory Address <sup>62</sup>	PCI Express Configuration Space
A[27:20]	Bus Number
A[19:15]	Device Number
A[14:12]	Function Number
A[11:8]	Extended Register Number
A[7:2]	Register Number
A[1:0]	Along with size of the access, used to generate Byte Enables

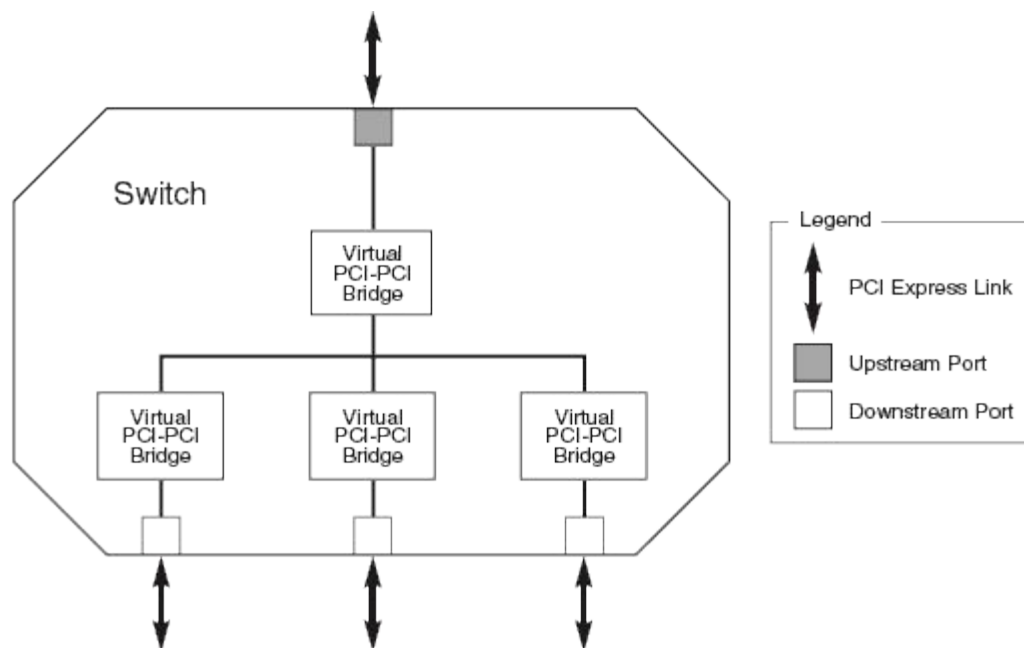
# Порт PCI Express

Порт – это логическая точка подключения соединения (Link), которая отвечает за управление линиями, сборку в пакеты исходящих данных и разборку входящих. Портами оснащены РС и коммутаторы (если они имеются). С точки зрения программирования порт представляет собой виртуальный мост PCI-PCI, а его Link – виртуальную подчиненную (вторичную) шину PCI.

Все порты делятся на корневые (принадлежат РС), нисходящие и восходящие (последние – только у коммутаторов).

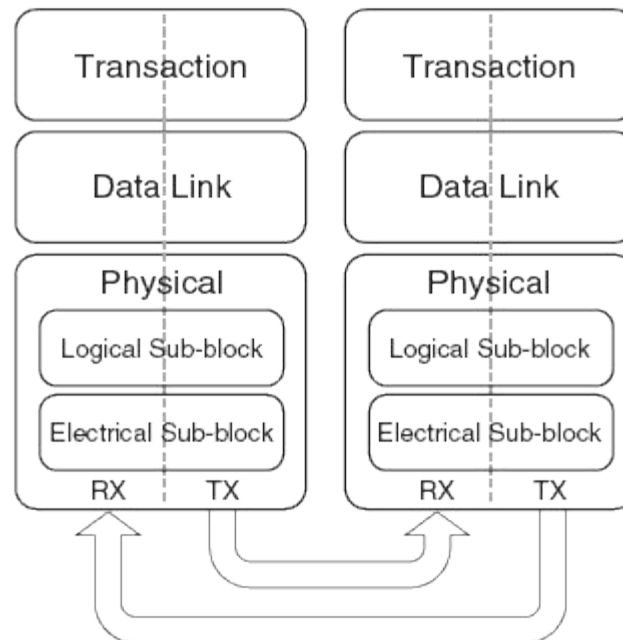
# Коммутатор PCI Express

Коммутатор служит для расширения количества подключаемых устройств, это аналог моста дополнительных шин PCI. Программно коммутатор представляет собой набор мостов PCI-PCI. Один из портов коммутатора ведет к порту PC или другого коммутатора.



# Уровни протокола PCI Express

В отличие от PCI протокол PCI Express условно разделен на уровни, без уточнения способов их реализации. Уровней всего три, на каждом выполняется сборка и разборка пакетов и их обрамление необходимыми заголовками и контрольными суммами. Не все пакеты относятся к уровню транзакций, существуют пакеты только канального уровня, служащие для управления.



# Уровень транзакций

Этот уровень отвечает в основном за выполнение операций чтения и записи в память либо в порты ввода-вывода.

Все транзакции, требующие ответа (обычно чтение), выполняются как расщепленные (Split): их инициатор получает статус запросчика (Requester), а целевое устройство – статус исполнителя (Completer).

На уровне транзакций поддерживается 4 адресных пространства:

- Памяти (основное)
- Портов в-в (для совместимости)
- Конфигурационное
- Пространство сообщений (Message Space)

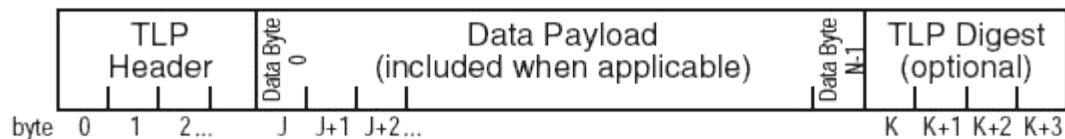
Последнее используется для эмуляции сигналов шины PCI (INTx#, PME# и др.) – т.н. «виртуальные провода».

# Пакеты уровня транзакций

Пакеты шины PCI Express оптимизированы для передачи по высокоскоростным последовательным линиям. Они имеют переменный формат, в том числе длину, чтобы исключить передачу незадействованных полей.

Первым передается наиболее значимый байт, обычно байт №0, чтобы приемное устройство могло начать его обработку до прихода остальных байтов.

Формат (обобщенный) пакета TLP следующий:



Длина пакета выровнена по границе dword. Код ECRC обеспечивает защиту инвариантных областей TLP.



TLP Type	Fmt [1:0] <sup>2</sup>	Type [4:0]	Description
MRd	00 01	0 0000	Memory Read Request
MRdLk	00 01	0 0001	Memory Read Request-Locked
MWr	10 11	0 0000	Memory Write Request
IORd	00	0 0010	I/O Read Request
IOWr	10	0 0010	I/O Write Request
CfgRd0	00	0 0100	Configuration Read Type 0
CfgWr0	10	0 0100	Configuration Write Type 0
CfgRd1	00	0 0101	Configuration Read Type 1
CfgWr1	10	0 0101	Configuration Write Type 1
Msg	01	1 0r <sub>2</sub> r <sub>1</sub> r <sub>0</sub>	Message Request – The sub-field r[2:0] specifies the Message routing mechanism (see Table 2-11).
MsgD	11	1 0r <sub>2</sub> r <sub>1</sub> r <sub>0</sub>	Message Request with data payload – The sub-field r[2:0] specifies the Message routing mechanism (see Table 2-11).
Cpl	00	0 1010	Completion without Data – Used for I/O and Configuration Write Completions and Read Completions (I/O, Configuration, or Memory) with Completion Status other than Successful Completion.
CplD	10	0 1010	Completion with Data – Used for Memory, I/O, and Configuration Read Completions.
CplLk	00	0 1011	Completion for Locked Memory Read without Data – Used only in error case.
CplDLk	10	0 1011	Completion for Locked Memory Read – otherwise like CplD.
			All encodings not shown above are Reserved.



# Форматы заголовков

Для запросов чтения памяти (запись не требует ответа):

	+0								+1								+2								+3							
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Byte 0 >	R	Fmt x 1			Type				R	TC		Reserved				T	E	Attr		R	Length											
Byte 4 >	Requester ID								Tag								Last DW BE				1st DW BE											
Byte 8 >	Address[63:32]																															
Byte 12 >	Address[31:2]																															R

Для запросов портов в-в:

	+0								+1								+2								+3							
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Byte 0 >	R	Fmt x 0			Type				R	TC 0 0 0		Reserved				T	E	Attr 0 0		R	Length 0 0 0 0 0 0 0 0 0 0 0 1											
Byte 4 >	Requester ID								Tag								Last DW BE 0 0 0 0				1st DW BE											
Byte 8 >	Address[31:2]																															R

# (продолжение)

Для запросов конфигурационных:

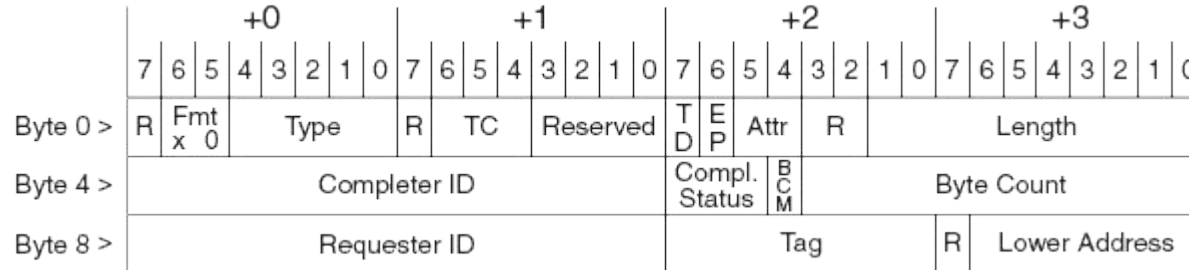
	+0								+1								+2								+3							
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Byte 0 >	R	Fmt x 0			Type				R	TC 0 0 0			Reserved				T	E	Attr 0 0		R	Length 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1										
Byte 4 >	Requester ID								Tag								Last DW BE 0 0 0 0				1st DW BE											
Byte 8 >	Bus Number								Device Number				Function Number				Reserved				Ext. Reg. Number				Register Number				R			

Для запросов типа Message:

	+0								+1								+2								+3							
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Byte 0 >	R	Fmt x 1			Type				R	TC			Reserved				T	E	Attr 0 0		R	Length										
Byte 4 >	Requester ID								Tag								Message Code															
Byte 8 >	{Fields in bytes 8 through 15 depend on type of Message}																															
Byte 12 >																																

# (продолжение)

Для ответов завершений:



Коды ответов:

Completion Status[2:0] Field Value	Completion Status
000b	Successful Completion (SC)
001b	Unsupported Request (UR)
010b	Configuration Request Retry Status (CRS)
100b	Completer Abort (CA)
all others	Reserved

# Использование сообщений

Сообщения могут применяться для различных управляющих целей.

Эмуляция прерываний INTx# выполняется с помощью посылки сообщения с кодом установки либо снятия одного из 4 флагов прерываний (INTA-INTD).

Эмуляция PME#, а также других состояний энергопотребления, включая события недостатка питания, также выполняется с помощью сообщений.

Сообщения об ошибках передают один из трех кодов: исправимая (Correctable), не фатальная (Non-fatal) и фатальная (Fatal) ошибка.

Есть также сообщения о событиях Hot-plug (индикаторы Power и Attention, кнопка отключения и т.п.), а также событиях, определенных производителем.

# Канальный уровень (Data Link Layer)

Отвечает за обеспечение целостности и достоверности данных, а также управление соединением.

На этом уровне пакеты уровня транзакций (TLP – Transaction Layer Packet) дополняются уникальным номером и контрольной суммой CRC.

Уровень проверяет порядок пакетов и контролирует их содержание, запрашивает пропущенные пакеты, сигнализирует о сбоях соединения, управляет состояниями соединения (неактивно, режим ожидания/инициализации, активно), служит для подачи сигналов энергопотребления, индикации ошибок и журналирования, обмена информацией управления потоком и т.д.

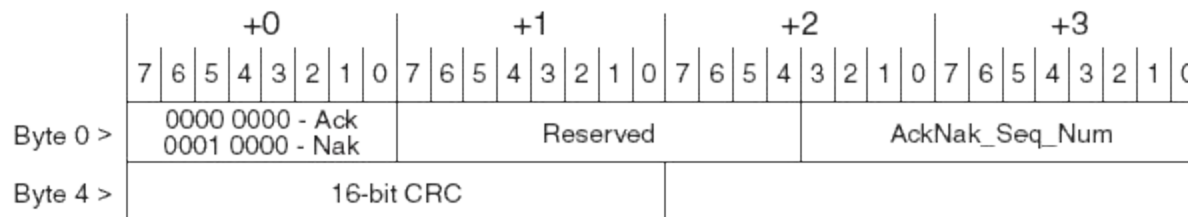
Специальные пакеты DLLP (Data Link Layer Packet) – служебные, данных не содержат, служат для управления соединением. Они не проходят через промежуточные узлы, распространяются только между портами.

# Пакеты DLLP

Подразделяются на следующие типы:

- Ack – подтверждение прихода TLP с заданным номером
- Nack – запрос на повтор TLP с заданным номером
- Пакеты управления кредитами и VC
- Пакеты управления PM

DLLP содержит заголовок с типом пакета, информационное поле и 16-битный CRC (LCRC).

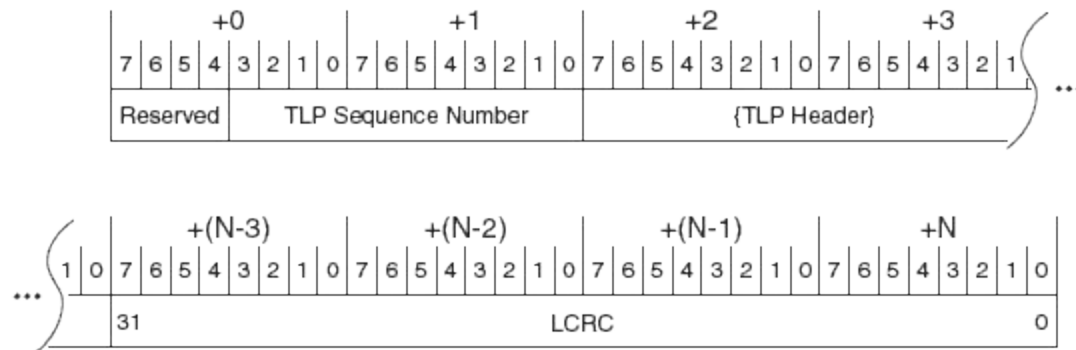


# Оборачивание TLP

Уровень канала сопровождает пакет TLP уникальным номером и 32-битным кодом LCRC (Link CRC). TLP находится в retry-буфере до прихода DLLP типа Ask с тем же номером.

Код LCRC работает только в пределах одного соединения.

Существуют развитые правила запроса и выполнения повторов, таймеров ожидания ответа (в зависимости от размера пакета и ширины линии) и т.д.

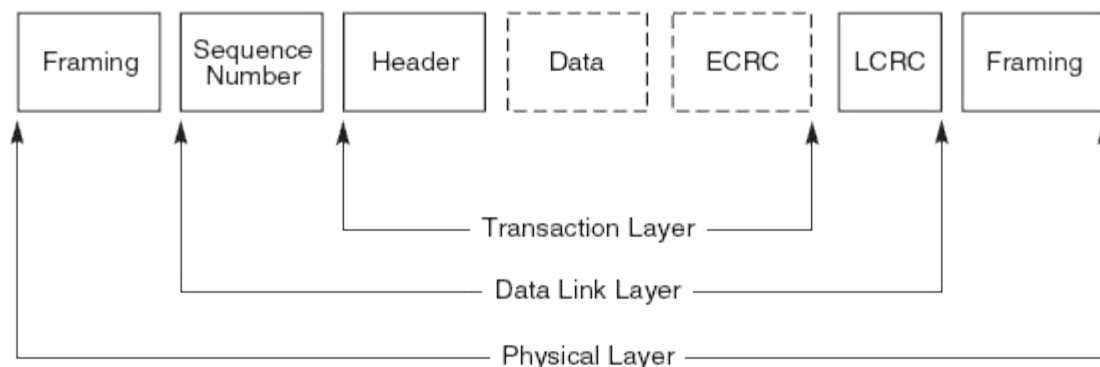


# Физический уровень

Делится на два подуровня – логический и собственно электрический.

На логическом уровне байты полученных данных кодируются по схеме 8b/10b и преобразуются в 10-битные символы. Выполняется также скрэмблирование (если необходимо), распределение по линиям, кадрирование, обрамление служебными символами.

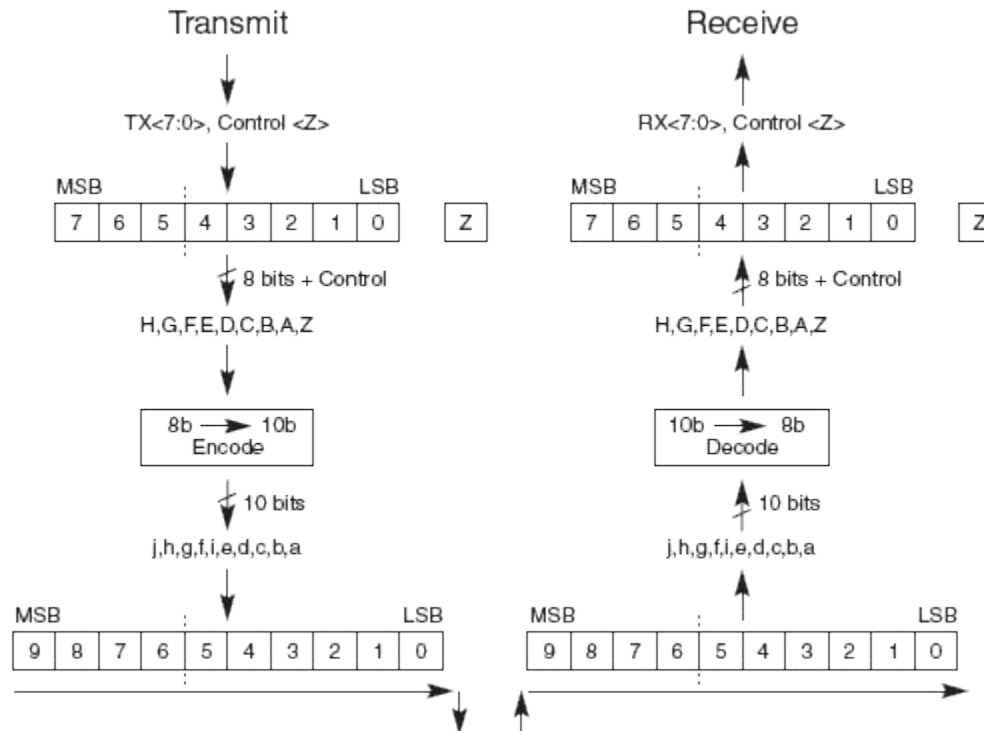
В результате данные принимают следующий вид:





# Кодирование 8b/10b

Кодирование 8b/10b выполняется по стандарту ANSI X3.230-1994 (или IEEE 802.3z). Младшие 5 бит отображаются на 6 бит, старшие 3 бита – на 4 бита, передаются младшим битом вперед



## (продолжение)

Специальные символы отделяют начало и конец TLP и DLLP, а также служат для калибровки, согласования скоростей портов, т.д.

При передаче по нескольким линиям начало TLP или DLLP передается только по линии №0.

Электрический суб-блок: две дифференциальные пары (D+ и D-), напряжение 0.1-0.8 В, нулевой уровень – 0.25 В, максимальная разность – 0.6 В.

Сигналы шины:

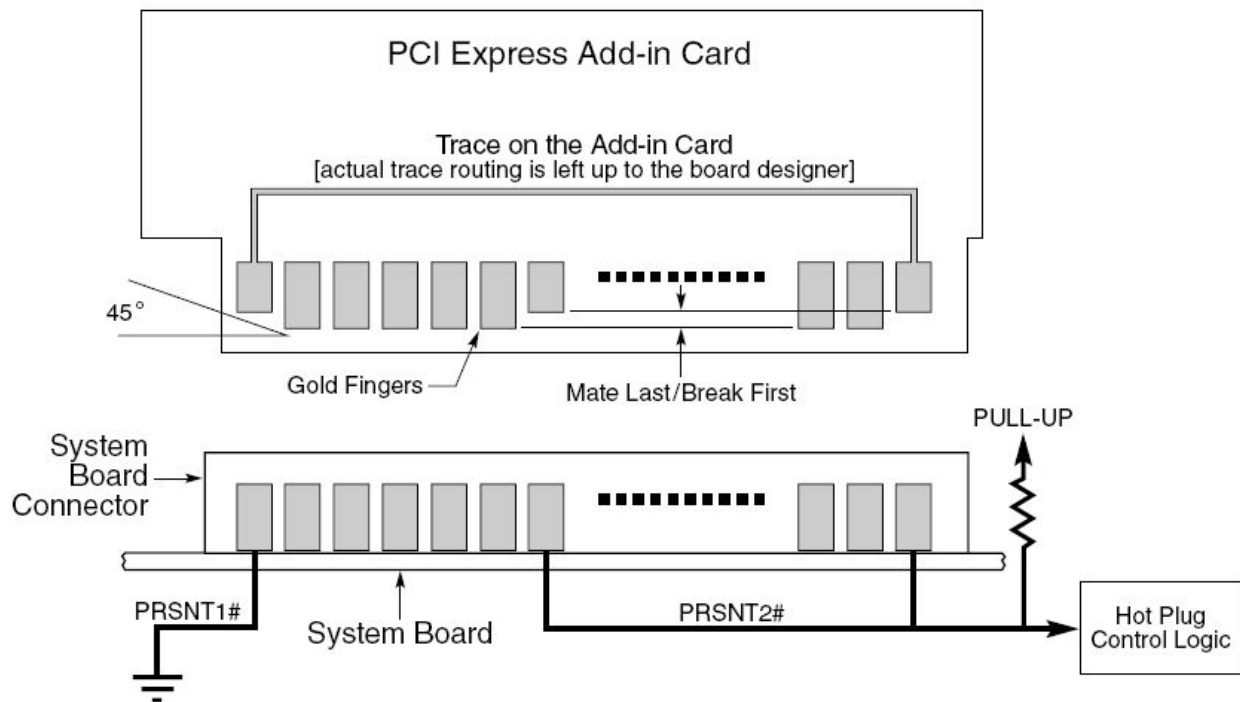
- PEr0, PEn0,.. PEr15, PEn15 – выходы передатчиков
- PERp0, PERn0,.. PERp15, PERn15 – выходы приемников
- REFCLK-, REFCLK+ - опорная частота 100 МГц
- PERST# - сброс карты
- WAKE# - пробуждение от карты

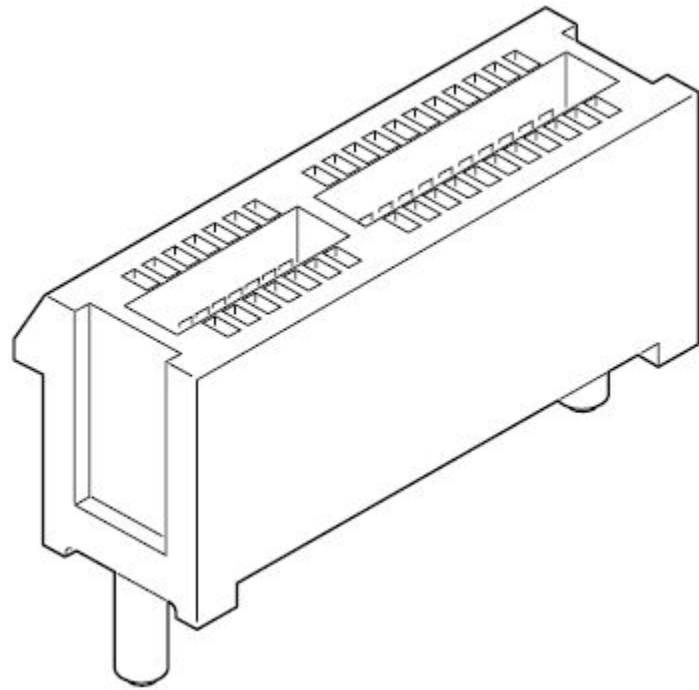
# Карта PCI Express

По линиям PRSNT1/PRSNT2 производится определение наличия карты.

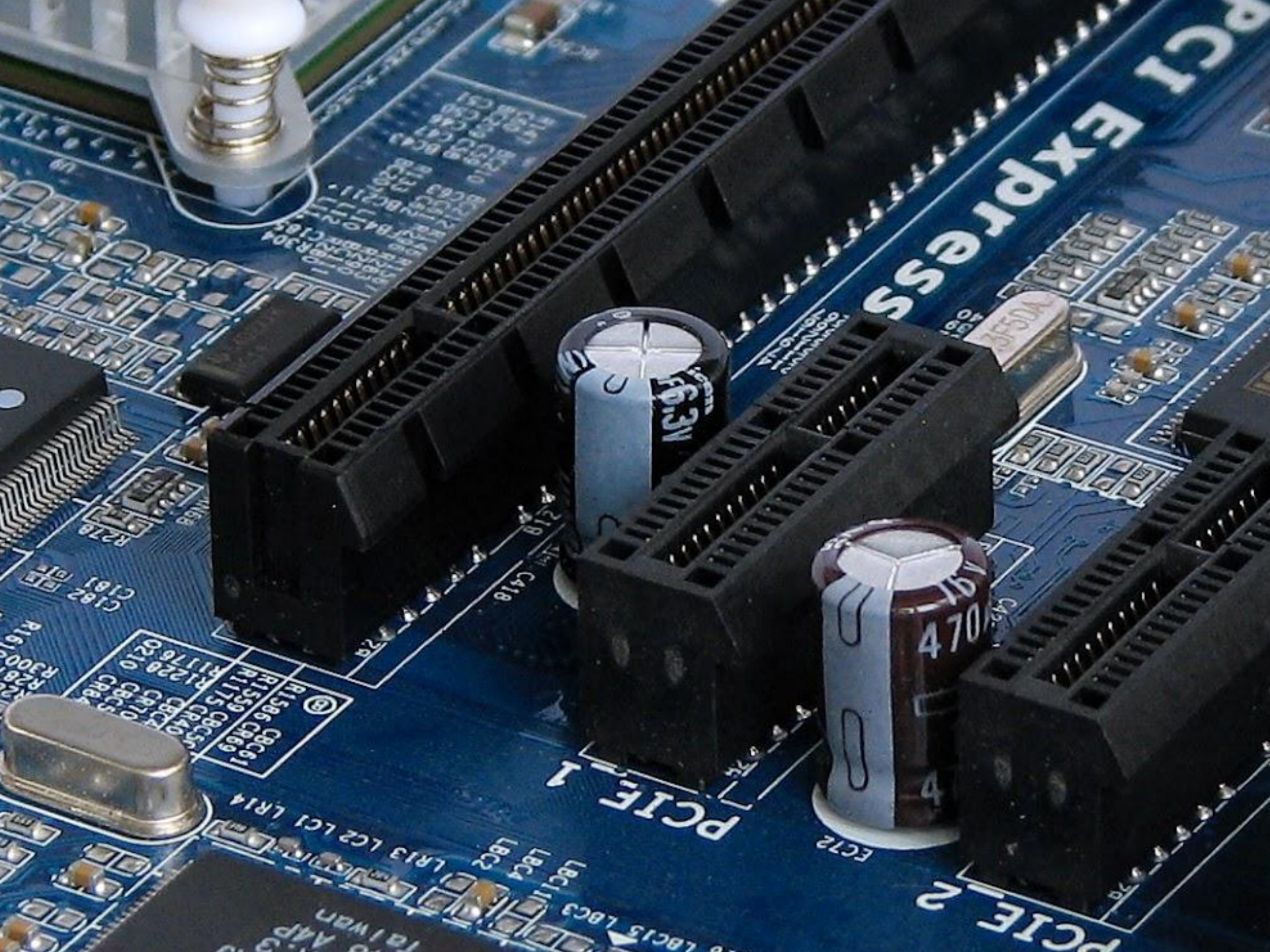
Для каждого формата слота линия PRSNT2 находится в последнем ряду, PRSNT1 – в первом.

Подается питание +12 В, +3.3 В, +3.3Vaux. Также в слоте разведены интерфейсы SMBus и JTAG.









# Карта PCI Express Mini Card

Специальный форм-фактор PCI Express Mini Card создан для карт расширения, устанавливаемых в мобильные компьютеры и мини-ПК.

Он предусматривает описание стандартных габаритов и разъема уменьшенного размера, а также дополнительных внешних выводов карты (антенна, светодиоды, сетевые розетки и т.д.).

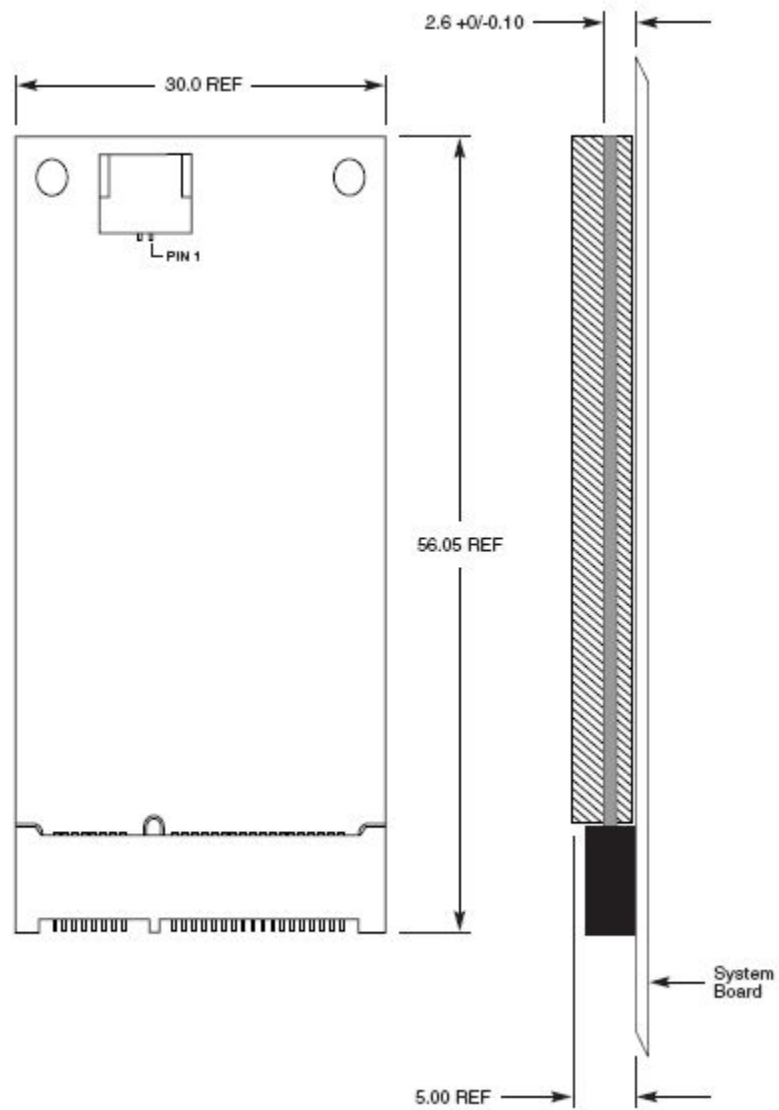
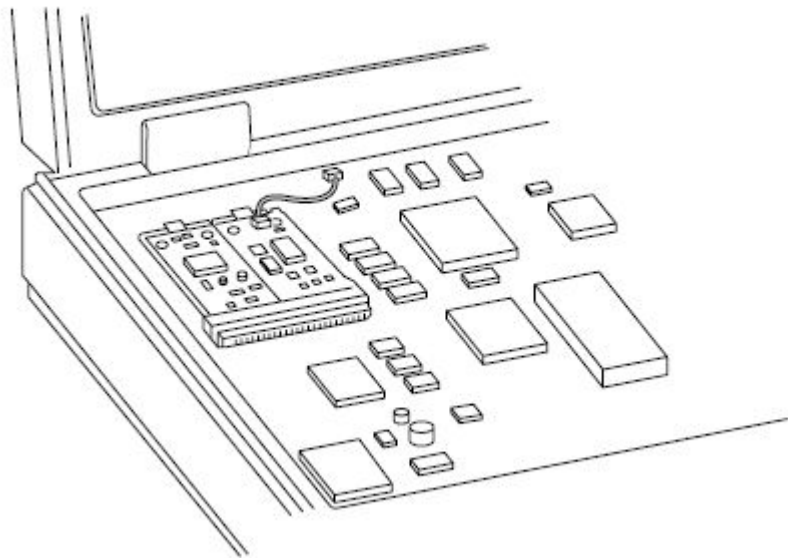
Основное назначение карт Mini Card – сетевые и коммуникационные устройства (адаптеры WiFi, WiMax, Bluetooth, GPRS/CDMA/UMTS), которые должны быть модульными и легко заменяемыми.

Речь не идет о пригодности к замене самим пользователем. Проблема в другом: существующие законодательные нормы использования радиочастотного диапазона не позволяют использовать все типы сетевых устройств в некоторых странах. Производитель ноутбука должен выбирать тип коммуникационной карты в зависимости от страны назначения.

Карта Mini Card реализует два интерфейса – системный PCI Express x1 и периферийный USB:

Signal Group	Signal	Direction	Description
Power	+3.3V (2 pins)		Primary 3.3 V source
	+3.3Vaux (1 pin)		Auxiliary 3.3 V source
	+1.5V (3 pins)		Primary 1.5 V source
	GND (12 pins)		Return current path
PCI Express	PETp0, PETn0 PERp0, PERn0	Input/Output	PCI Express x1 data interface: one differential transmit pair and one differential receive pair
	REFCLK+, REFCLK-	Input	PCI Express differential reference clock (100 MHz)
Universal Serial Bus (USB)	USB_D+, USB_D-	Input/Output	USB serial data interface compliant to the USB 2.0 specification
Auxiliary Signals (3.3V Compliant)	PERST#	Input	Functional reset to the card
	CLKREQ#	Output	Reference clock request signal
	WAKE#	Output	Open Drain active Low signal. This signal is used to request that the system return from a sleep/suspended state to service a function initiated wake event.
	SMB_DATA	Input/Output	SMBus data signal compliant to the SMBUS 2.0 specification
	SMB_CLK	Input	SMBus clock signal compliant to the SMBUS 2.0 specification
Communications Specific Signals	LED_WPAN#, LED_WLAN#, LED_WWAN#	Output	Active low signals. These signals are used to allow the PCI Express Mini Card add-in card to provide status indicators via LED devices that will be provided by the system.







# Карты ExpressCard

Организация PCMCIA, занимающаяся формализацией разработок в области карт расширения для ноутбуков с «горячим» подключением, предложила новый стандарт карт расширения – ExpressCard. От стандарта PC Card он унаследовал только некоторые из габаритов корпуса и общую конструкцию.

Фактически в корпусе модуля ExpressCard может быть помещено устройство с интерфейсом либо PCI Express x1, либо USB. В версии ExpressCard 2.0 обеспечена поддержка PCI Express 2.0 и USB 3.0, что позволяет устройствам получить канал с пропускной способностью 5 Гбит/с – достаточно для внешних винчестеров, ТВ-тюнеров, широкополосных модемов, виртуальных видеокарт и других требовательных устройств.

Функции управления энергопотреблением уже встроены в PCI Express и особенно USB, что сокращает стоимость внедрения ExpressCard.

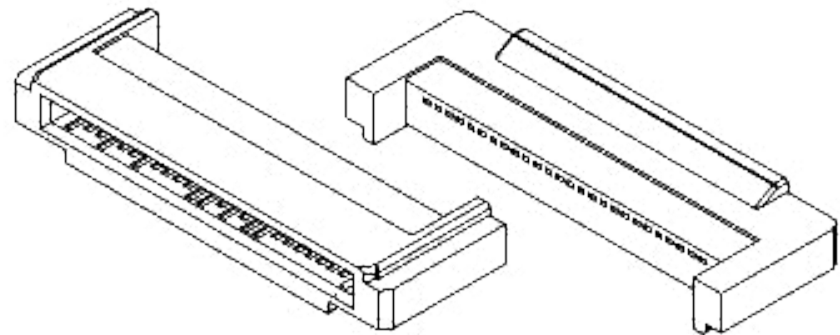
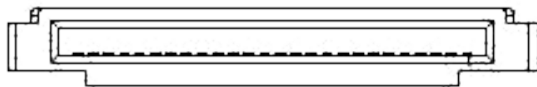
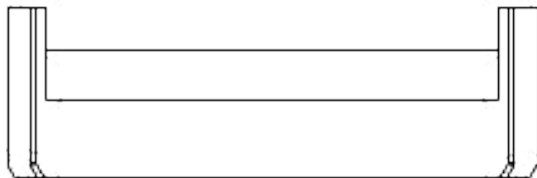
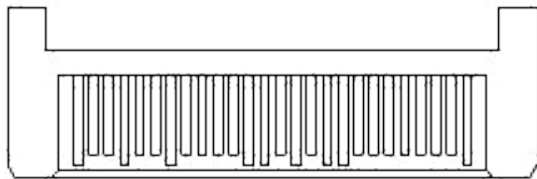
# Физический интерфейс

По сути ExpressCard описывает только физический интерфейс – размер модуля и формат разъемов.

Благодаря тому, что интерфейсы PCI Express и USB последовательные, удалось сократить размеры разъема (по сравнению с PC Card) и реализовать сразу два интерфейса.

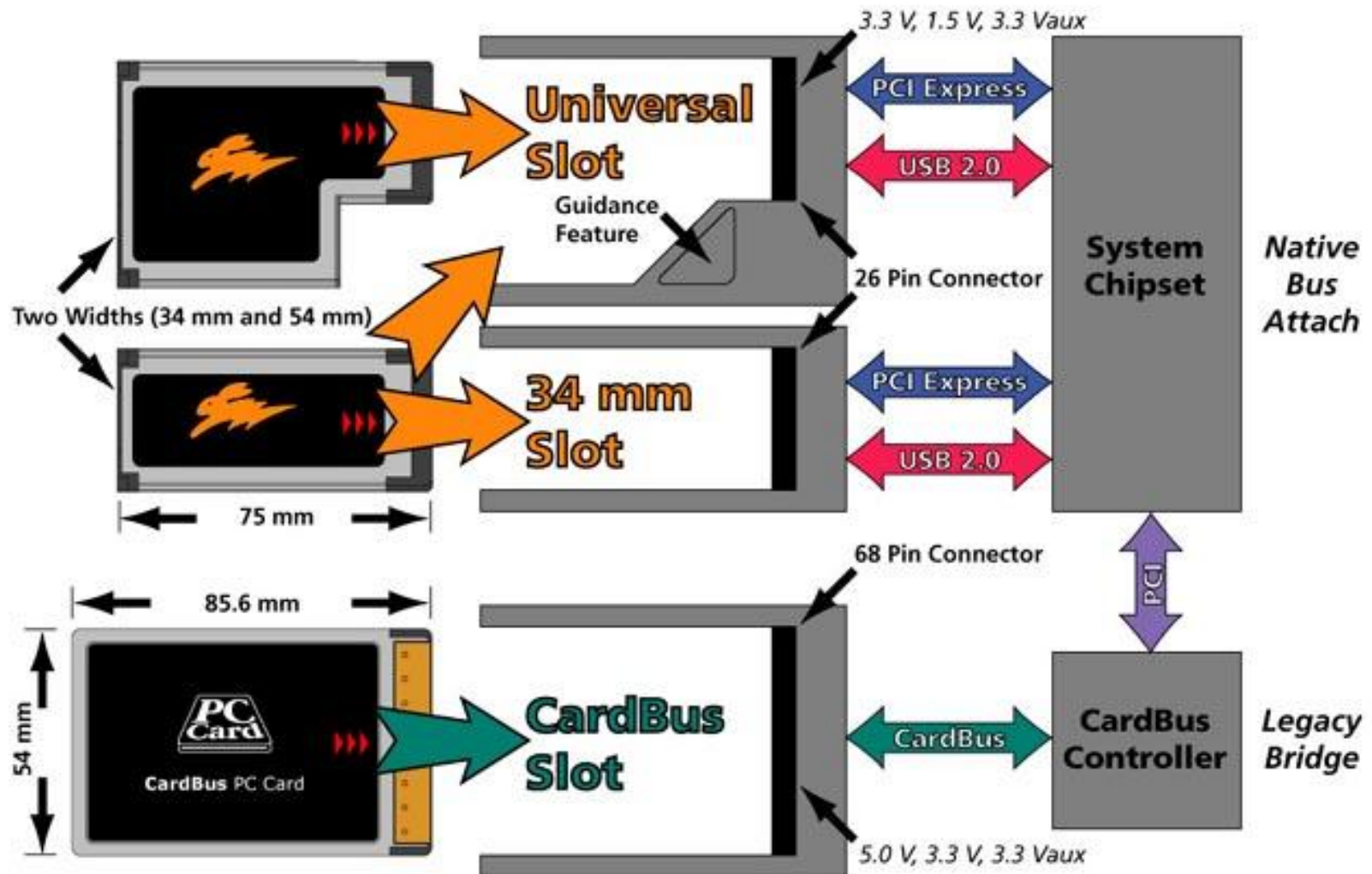
Карты ExpressCard имеют единую толщину (5 мм) и различаются только шириной – 34 мм или 54 мм (для устройств, которые не помещаются в корпус 34 мм), разъем идентичен. Слоты могут быть универсальными или только для устройств 34 мм.

# Разъемы ExpressCard

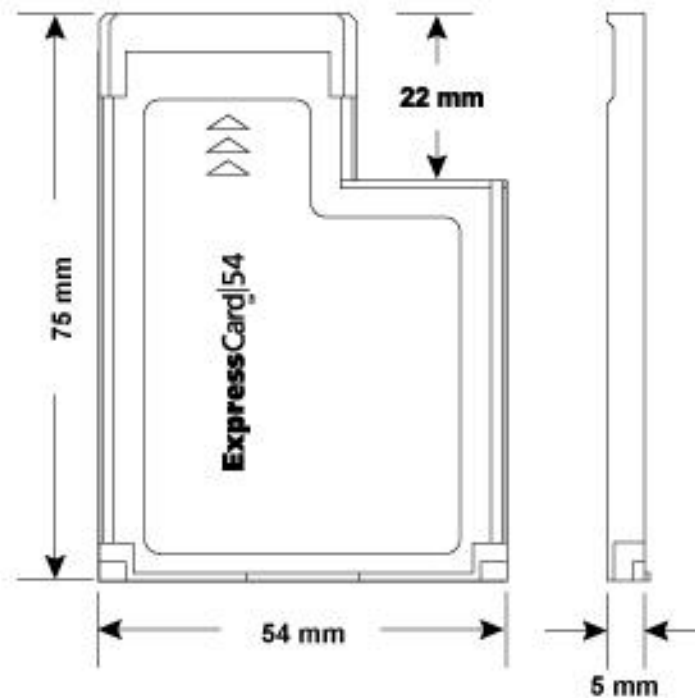
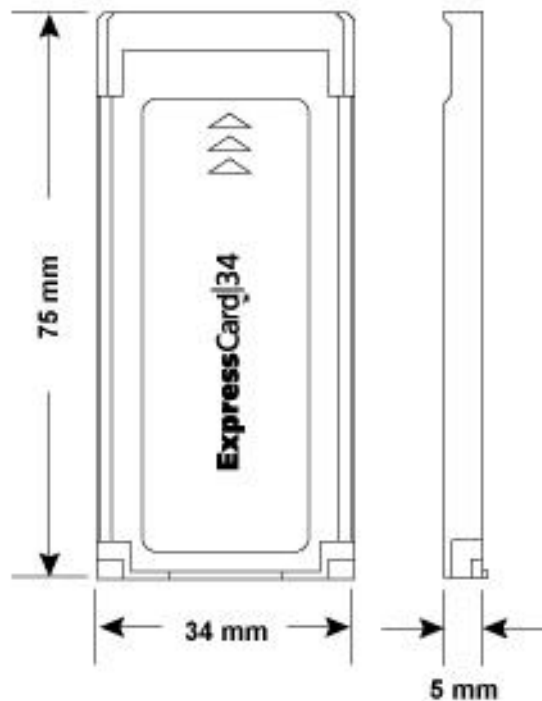


# Модули ExpressCard

## ExpressCard Technology vs. CardBus



# Модули ExpressCard



Signal Group	Signal	Description	Interface Type(s) on module			
			PCI Express	USB	Both	Host
PCI Express	PETp0, PETn0, PERp0, PERn0	PCI Express x1 interface	R	NC	R	R
	REFCLK+, REFCLK-	PCI Express reference clock	R	NC	R	R
	PERST#	PCIe Reset	R	NC	R	R
USB	USBD+, USBD-	USB serial data interface	NC	R	R	R
SMBus	SMBDATA, SMBCLK	SMBus	Opt	Opt	Opt	Opt
System	CPPE#	PCI Express interface presence detect	R	NC	R	R
Auxilliary	CLKREQ#	Request that REFCLK be enabled	R	NC	R	Opt
Signals	WAKE#	Request that the host interface return to full operation and respond to PCI Express	Opt	NC	Opt	Opt
	CPUSB#	USB interface presence detect	NC	R	R	R
Power	+3.3 V (2 Pin)	Primary voltage source, 3.3V	R	R	R	R
	+3.3 VAUX (1 Pin)	Auxiliary voltage source, 3.3VAUX	Opt	Opt	Opt	R
	+1.5 V (2 Pins)	Secondary voltage source, 1.5V	Opt	Opt	Opt	R
	GND (4 Pins)	Return current path, Ground	R	R	R	R