Что скрывает Intel Xeon?

31 Мар 2016

Илья Манусов

Пользователям Linux хорошо знакомы утилиты получения системной информации *lscpu* (данные о процессоре), *lspci* (данные о конфигурационном пространстве PCI) и подобные. Как оказалось, несмотря на рутинный статус, утилита *lspci* позволяет узнать о наличии системных ресурсов, размещенных в адресном пространстве с некоторым отклонением от принятого стандарта.

Платформа и инструменты

Отход от рекомендаций, принятых и одобренных разработчиками PCI-спецификации, давно стал «*плодотворной дебютной идеей*»© и для энтузиастов, и для злоумышленников. В настоящей статье мы продолжим дискурсагрессивного сканирования шины PCI, с той разницей, что данная исследуемая платформа построена на процессоре Intel Xeon E5-1650 (микроархитектура Sandy Bridge), и нашей целью является то, что производитель называет Intel Confidential, а конкретнее — секреты этого процессора.

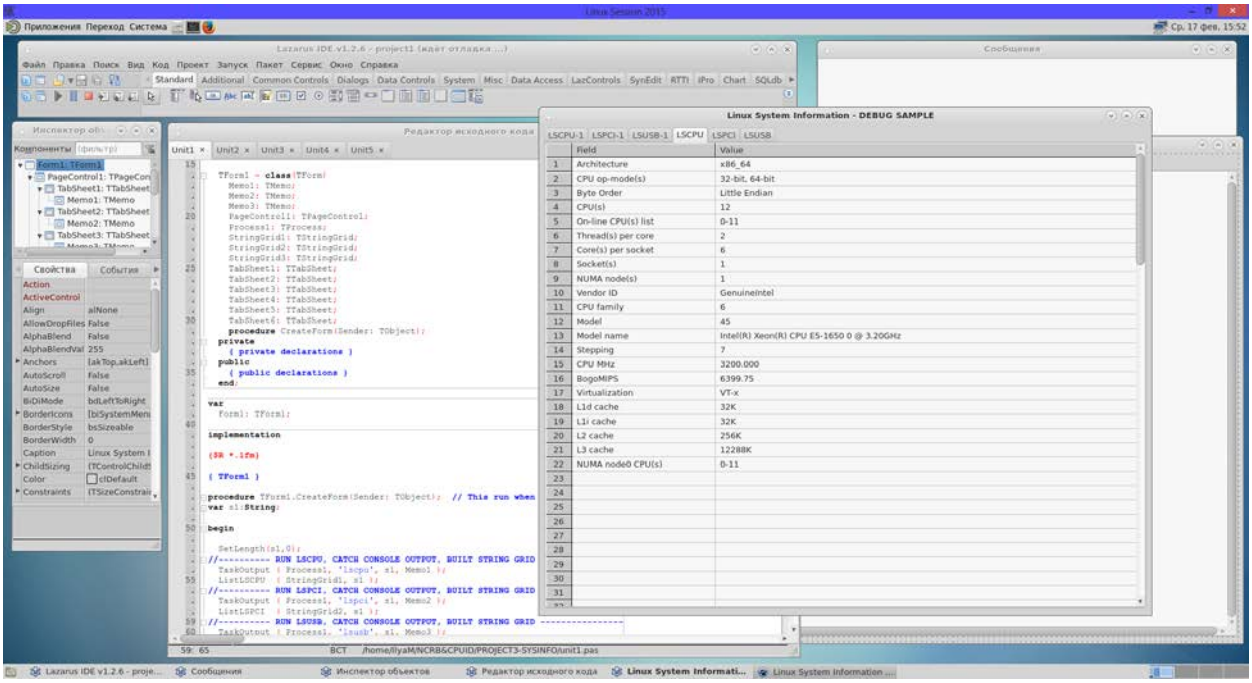

Рис.1. *Результат работы штатной Linux-утилиты lscpu, программно сохраненный и визуализируемый в таблице TStringGrid: для создания оболочки используется Free Pascal и среда разработки Lazarus*

В таких случаях говорят: «ничто не предвещало беды». Задача, при решении которой неожиданно открылась описываемая особенность, состояла в написании небольшой программной оболочки, запускающей штатные Linux-утилиты *lscpu, lspci*, *lsusb*, сохраняющей выходной поток этих консольных приложений и визуализирующей полученные данные в виде GUI-таблиц.

Как выполняется сканирование PCI

Допустим, операционная система или информационно-диагностическая утилита, желает получить список ресурсов, адресуемых в конфигурационном пространстве. Как известно, для этого нужно выполнить *сканирование (PCI bus scan)*. Оно состоит в последовательном чтении и анализе содержимого регистров по заданным адресам Bus, Device, Function. Отметим, что номер PCI-шины (параметр Bus) имеет разрядность 8 бит, следовательно, в системе может быть максимум 256 шин, номера которых находятся в интервале *ooh-FFh* (в шестнадцатеричной системе). В целях сокращения времени сканирования, а также корректного разграничения адресных диапазонов,*Legacy BIOS*, либо *UEFI Firmware* декларирует максимальный используемый номер шины. Согласно стандарту, сканирование шин, с номерами больше максимального выполнять не следует.

В чем отклонение от стандарта

Читатель уже видимо догадался, что на исследуемой платформе обнаружены ресурсы, адресуемые на шинах с номерами выше максимального. Обратимся к скрин-шотам (Рис.2 и Рис.3). В этом примере, «законопослушные» устройства подключены к шинам с номерами 0-7, «скрытые» ресурсы находятся на шине с адресом FFh.

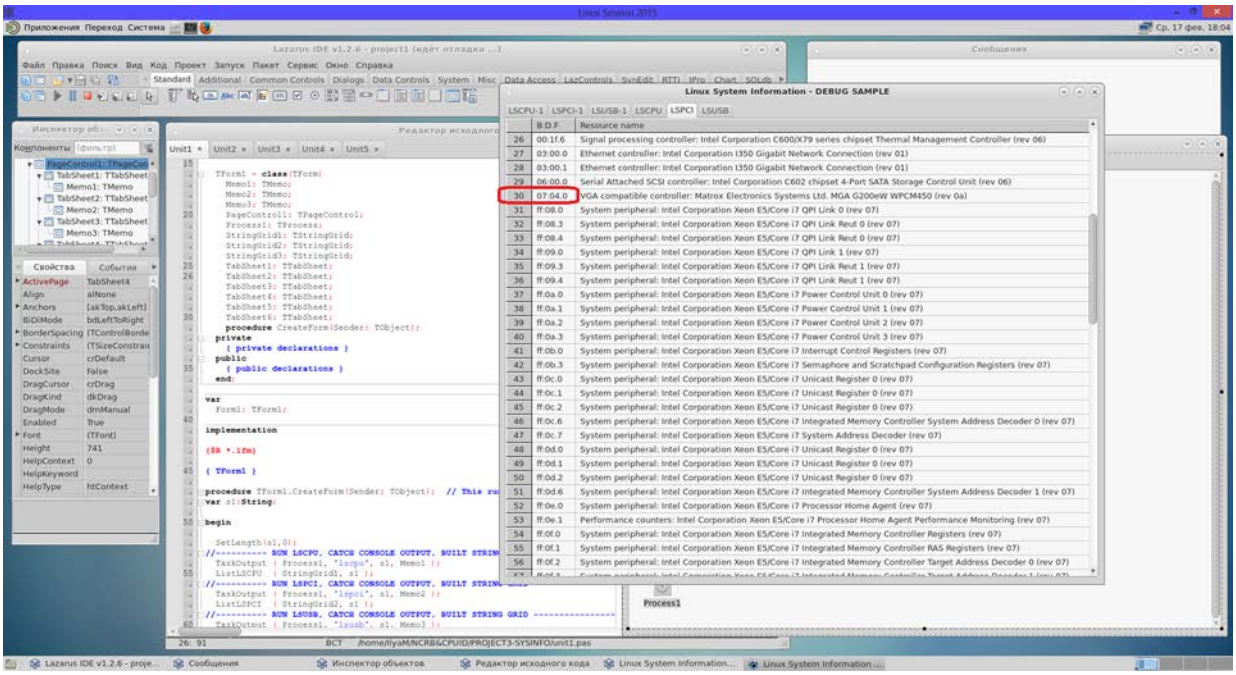

Рис.2. *Результат работы штатной Linux-утилиты lspci, программно сохраненный и визуализируемый в таблицеTStringGrid: видим ресурсы, адресуемые на PCI-шине с номером 255 (FFh); последний «законный» номер шины (07), в данном примере используется видео адаптером, его адрес Bus, Device, Function выделен красным контуром*

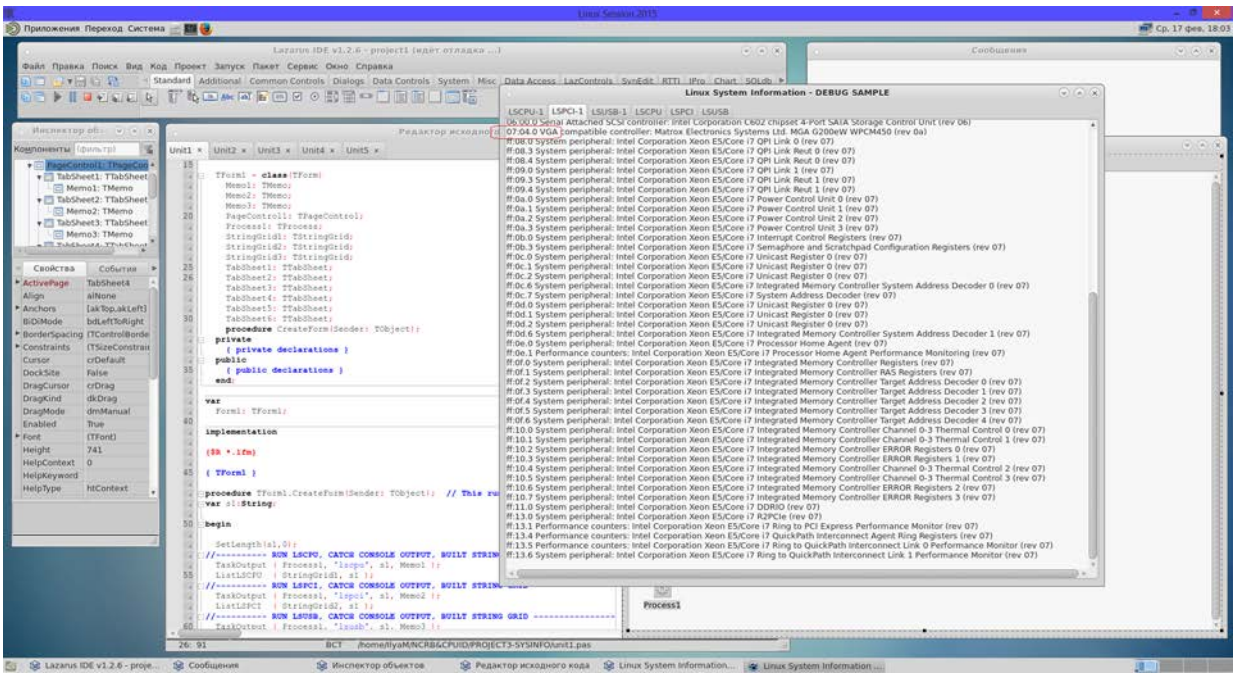

Рис.3. *Та же информация, сохраненная и визуализируемая в таблице TMemo: менее эстетично, но визуализируется больше строк*

Подробнее о скрытых ресурсах

Функциональное назначение обнаруженных регистровых блоков с различной степенью подробности описано в документации Intel. Принципиально то, что наличие данных ресурсов в адресном пространстве, а также номер шины, используемый при доступе к ним, являются программно конфигурируемыми опциями. Это означает, что*ответственность за отклонение от спецификации PCI возлагается на разработчиков Firmware*, также как это имело место в ранее рассмотренном примере с агрессивным сканированием шины PCI, ссылка на который приведена в начале статьи.

Figure 1-2. Processor Uncore Devices Map

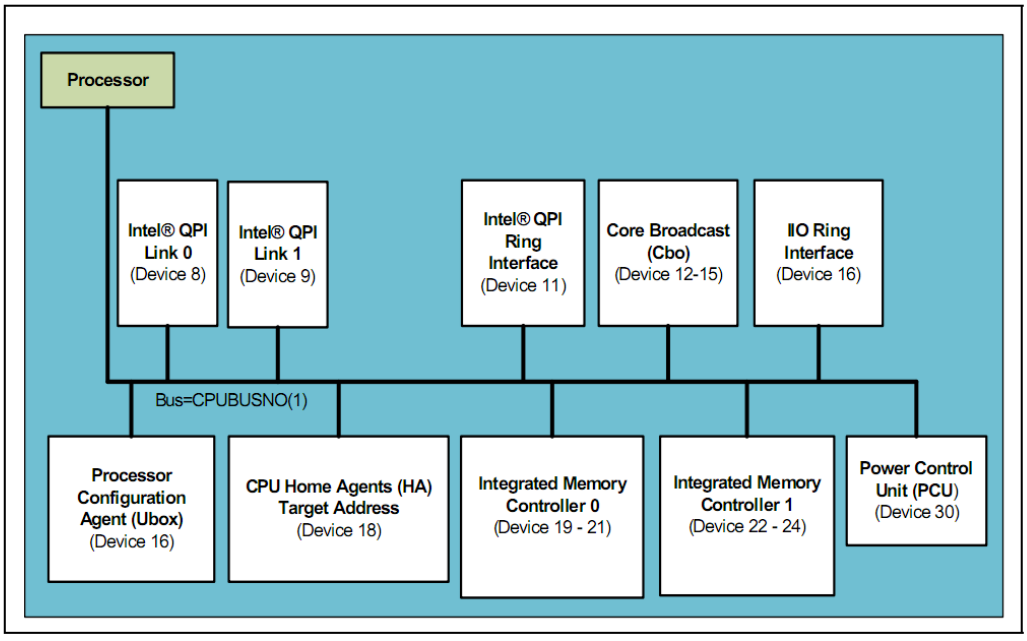

Рис.4. *Блок-схема подключения «скрытых» системных ресурсов, согласно документации Intel Xeon Processor E5-1600 and E5-2600 v3 Product Families, Volume 2 of 2, Registers*

Как видно из блок-схемы, большинство рассматриваемых компонентов отвечают за интерфейс процессора с внешним миром, в частности, шины *QPI (Quick Path Interconnect)* обеспечивают взаимодействие между процессорами мультипроцессорной системы, а также между процессором и системной логикой. Компонент*Integrated Memory Controller* обеспечивает доступ к конфигурационным регистрам контроллера оперативной памяти. *Power Control Unit (PCU)* управляет параметрами электропитания и термоконтроля. Компонент *IIO(Integrated Input/Output)* управляет доступом к периферийным ресурсам и контроллерам прерываний, поддержкой виртуализации и расширенной диагностикой ошибок.

3.1.1 QPIMISCSTAT: Intel QPI Misc Status

This is a status register for Common logic in Intel QPI. It is shared between Intel QPI 0 and Intel QPI 1.

QPIMISCSTAT Bus: 1		Device: 8	Function: 0 Offset: D4
Bit	**Attr**	**Default**	**Description**
2:0	RO-V	011b	**Intel QPI Rate** This reflects the current Intel QPI rate setting into the PLL. 011 - 6.4 GT/s 101 - 8 GT/s 111 - 9.6 GT/s other - Reserved

Рис.5. *Один из регистров позволяет прочитать текущую установленную скорость интерфейса QPI*

Примечание.

Отметим, что номер шины в таблице указан как константа Bus=1. Такой вариант — частный случай, номер шины является реконфигурируемым параметром.

Вместо послесловия

Как было показано выше, рассмотренные особенности адресации системных ресурсов лежат на совести разработчиков UEFI BIOS, так как именно низкоуровневое программное обеспечение при старте платформы назначает номер шины для их регистровых блоков. Максимальный доступный номер шины в системе, тем более в компетенции Firmware. Поэтому, именно оно в ответе за то, что первый параметр оказался больше второго.

А в качестве следующего опыта, попробуем осуществить доступ к регистрам устройств на шине с номером FFh, в среде UEFI. Для этого, можно, например, воспользоваться утилитой *RU32.EFI*. Только так может быть получено строгое доказательство физической доступности в адресном пространстве всех объектов, о которых сообщила нам утилита *lspci*.

Теги: Intel UEFI BIOS