



Процессор CELL

Часть 2. Модели программирования

Константин Недоводеев,
к.т.н., с.н.с. Института ВКиСТ, СПбГУАП

Санкт-Петербургский Государственный Политехнический Университет



Отличительные особенности

Наибольшую **трудность** при программировании процессора CELL вызывает **наличие** у SPE ядер **локальной памяти**. Однако, именно локальная память позволяет достичь высокой производительности.

SPE ядра поддерживают параллелизм по данным (**SIMD**).

В состав процессора входит PPE архитектуры Power, который способен выполнять код ОС, а также программы для процессоров Power без модификации. Но **для достижения максимальной производительности необходимо использовать SPE ядра!**



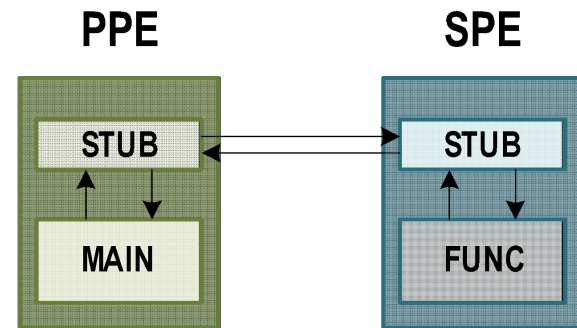
Рассматриваемые модели программирования

1. Модель **выгрузки функций** (function offload model);
2. Модель **ускорения вычислений** (computational acceleration model);
3. **Потоковые** модели (streaming models);
4. Модель **мультипроцессора с разделяемой памятью** (shared-memory multiprocessor model);
5. Модель **ассиметричных потоков** (asymmetric thread runtime model).

Модель выгрузки функций

Наиболее **проста** в реализации.

PPE выполняет основную программу.



SPE выполняет код функций, критичных к времени выполнения.

Программист **явным образом** указывает – какие функции должны выполняться на SPE.



Модель ускорения вычислений

Является развитием предыдущей модели.

Более гибкая.

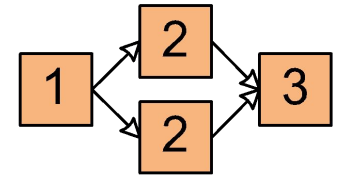
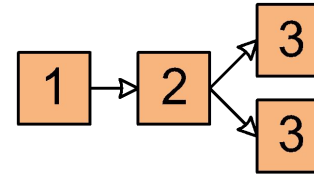
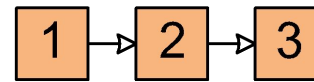
SPE выполняют наиболее вычислительно емкие участки кода программы.

Необходимо распараллеливать программу и данные.

Необходимо организовывать эффективный DMA обмен кодом и данными.



Потоковые модели



SPE ядра организуются в **вычислительный конвейер**.

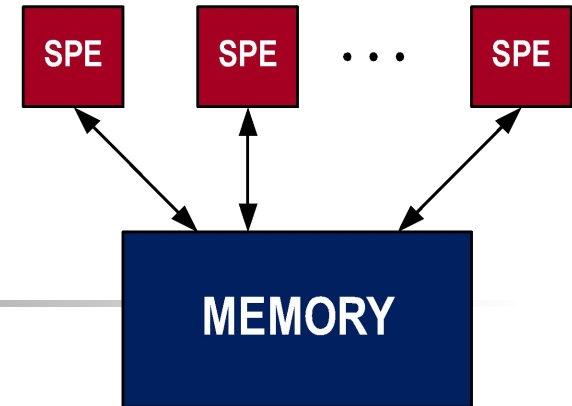
PPE играет роль контроллера потока (stream controller).

Конвейеры могут иметь различную конфигурацию.

Необходимо **равномерно загружать** SPE ядра для более полной загрузки.

SPE обмениваются сообщениями с PPE и между собой.

Модель мультипроцессора с разделяемой памятью



CELL можно программировать как мультипроцессор с разделяемой памятью.

Отличие от обычного мультипроцессора состоит в том, что операции загрузки/сохранения для разделяемой памяти заменяются на **пары** операций загрузки/сохранения для **локальной памяти** SPE и DMA операции для **разделяемой памяти**.

В локальной памяти SPE можно организовать **совмещенный программный кэш** инструкций и данных.

Eichenberger A.E., et. al. Optimizing compiler for a Cell processor // Proceedings of the PACT'05. – Washington: IEEE, 2005. – P. 161-172 (Compiler-Controlled Software Cache)



Модель асимметричных потоков

Является **расширением** широко распространенной модели многопоточности на случай наличия ядер с **различной архитектурой** инструкций.

SPE **не поддерживает** аппаратную смену контекста, поэтому на SPE **нельзя** организовать **вытесняющую многозадачность** (preemptive multitasking) Для SPE возможно применять **модель работы потока до завершения** (run-to-completion model), а также **кооперативную многозадачность** (cooperative multitasking).

Кооперативная многозадачность – каждый поток сам определяет момент времени передачи ресурсов другому потоку ОС.



Выводы

При программировании процессора CELL можно выбрать **любую** из вышеописанных моделей программирования. У каждой из моделей есть свои сильные и слабые стороны. Такой широкий выбор моделей программирования процессора CELL обусловлен:

1. наличием вычислительных ядер с **различной** архитектурой инструкций;
2. наличием **локальной памяти** у SPE ядер;
3. наличием **разделяемой памяти**;
4. присутствием в архитектуре процессора PPE **ядра архитектуры Power**;
5. а также, наличием **большого числа** вычислительных ядер.



Литература

1. Dominic Mallison, Mark DeLoura CELL: a New Platform for Digital Entertainment (Sony Computer Entertainment US Research and Development) <http://www.research.scea.com> 9.03.05
2. J.A. Kahle, M.N. Day et. al. Introduction to the Cell multiprocessor // IBM Journal of Research and Development Vol. 49 No. 4/5 July/September 2005