



Проектирование и исследование предсказателя переходов

Болдов Олег Евгеньевич

Программа магистратуры, для поступления на которую подается заявка

09.04.01 Информатика и вычислительная техника

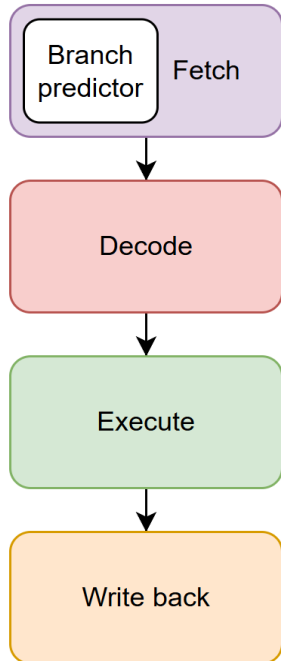


Рисунок 1 – Конвейер

Предсказатель переходов - модуль ядра, который значительно увеличивает производительность, так как позволяет делать загрузку из кэша инструкций (медленную операцию) заранее.

Однако нет стандартного способа выбора алгоритма и конфигурации предсказателя. Тратится много времени и ресурсов на разработку нескольких модулей, их тестирование и анализ.

Такой выбор часто упрощается из-за экономии, что приводит к неэффективным предсказателям. Данное исследование нацелено на разработку метода ускорения выбора предсказателя переходов, а также подбора эффективных структур по параметрам конвейера.



Рисунок 2 – Потеря из-за ошибки предсказателя

Цель: сформировать метод ускорения выбора предсказателя переходов под конкретные параметры системы.



Задачи:

- 1) Реализовать алгоритмы на языке system verilog
- 2) Получить трассы для тестирования, содержащие адрес jump-инструкции и «правильный» ответ (taken or not taken)
- 3) Реализовать тестовое окружение для запуска трасс и подсчёта статистики
- 4) Проанализировать результаты и сделать выводы в рамках разных моделей процессоров
- 5) Провести экспериментальное подтверждение работы метода

Алгоритмы предсказания

Static
всегда делаем
одинаковый выбор

bim - bimodal counter
Для каждого адреса будем хранить
счётчик. Считаем, taken или not taken
было больше за последние предсказания

GH
Global

PH
Path

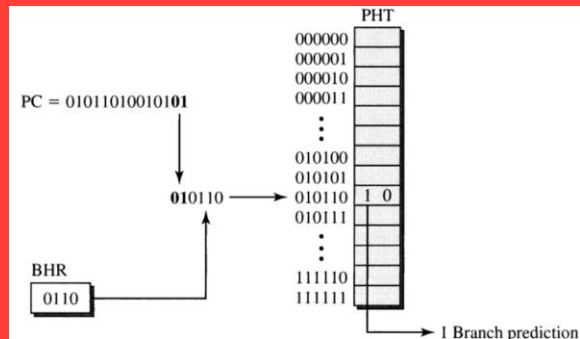


Рисунок 3 – двухуровневый предсказатель

GSHARE - $PC \wedge GH \wedge PH$

TAGE - TAgged GEometric
Для разных случаев нужны разные
длины историй. Воспользуемся
большим количеством таблиц,
выбирая тех, кто чаще выдает верные
решения

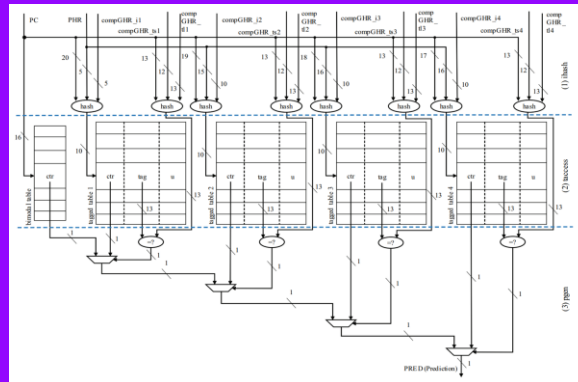


Рисунок 4 – предсказатель TAGE

Таблица 1 – Точность и память в изучаемых алгоритмах

	static	last	bim	gh	ph	gshare	tage
accuracy	61%	56%	68%	77%	79%	88%	95%
memory	1	1	8 Кб	8 Кб + 4	24 Кб	24 Кб + 4	>100 Кб
acc/mem	61	56	0,008	0,0094	0,0032	0,0036	0,0009

Чем больше требует памяти алгоритм, тем больше он может дать процент успеха, но тем дороже становится его получение

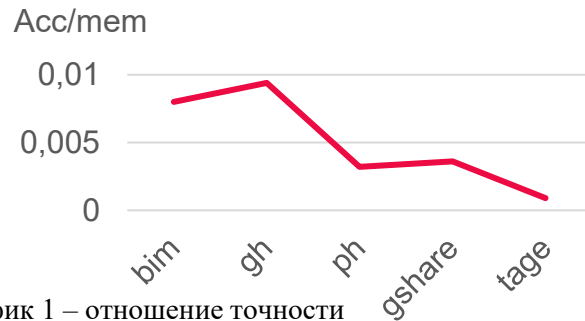


График 1 – отношение точности к затраченной памяти в разных предсказателях

Він зависимость от памяти

Таблица 2 – Точность и кол-во занятых ячеек в алгоритме «бимодальные счётчики» с разным размером памяти

ячейки	2^4	2^6	2^8	2^{10}	2^{12}	2^{14}	2^{16}
точность	54.6%	65.6%	71.6%	73.0%	73.2%	73.3%	73.3%
Занятые	8	32	128	440	777	891	916

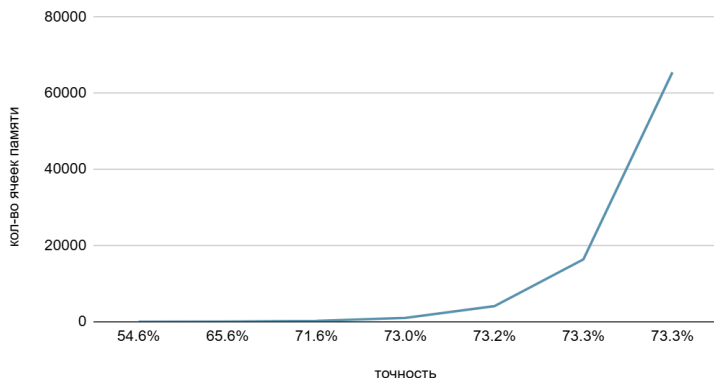


График 2 – Зависимость используемой памяти от точности

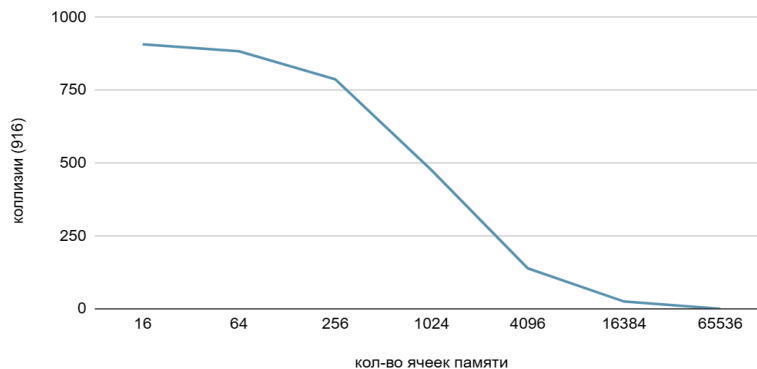


График 3 – Зависимость коллизий от используемой памяти

Штраф за очистку:

$$F = s \times d, \quad (1)$$

где s — ширина конвейера, d — длина конвейера — сколько тактов в среднем выполняется инструкция.

Кол-во инструкций, которые вызовут очистку:

$$J \times (100\% - A), \quad (2)$$

где J — среднее кол-во инструкций ветвления в задачах, A — точность, которую необходимо получить от предсказателя переходов.

Кол-во тактов, которые процессор потеряет из-за очистки:

$$J \times (100\% - A) \times F \quad (3)$$

↓

$$J \times (100\% - A) \times F + t_0 \leq t \quad (4)$$

где t_0 — время, за которое выполняется задача без очистки, t — максимальное время выполнения задачи, которое удовлетворяет запрос к производительности процессора.

Итоговое выражение для определения необходимой точности ПП:

$$A \geq 1 - \frac{(t - t_0)}{J \times d \times s}. \quad (5)$$



- ❑ В случаях, где предсказание сделать тяжело, стоит использовать статический предсказатель переходов
- ❑ При большой связности инструкций ветвления стоит добавить GH
- ❑ PH не стоит использовать без gshare
- ❑ Оптимальным количество бит индексации является 12 бит
- ❑ В случаях высокой вероятности большого количества коллизий допустимо увеличении количества бит индексации
- ❑ Не используя gshare, оптимальным количеством бит для длины историй является 4 бита
- ❑ В случаях с сильно связанным ветвлением допустимо увеличение длины истории
- ❑ Если в ходе использования метода была получена оценка, не удовлетворяющая целевую точность, необходимо использовать таблицу точностей, как пропорциональную полученной

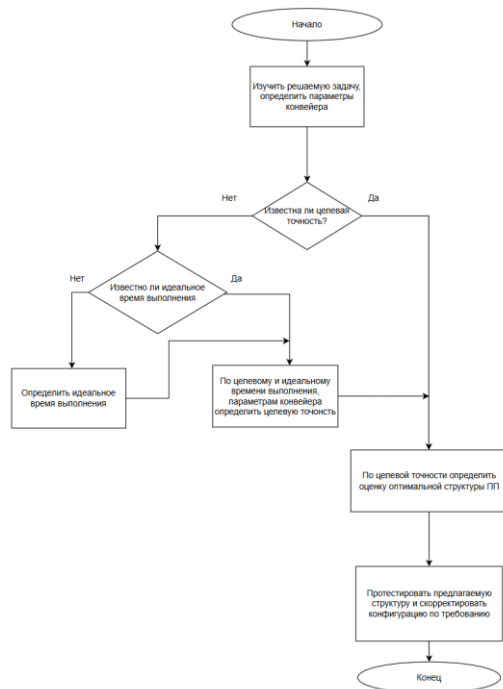


Рисунок 5 – Метод выбора структуры

$$A \geq 1 - \frac{(t - t_0)}{J \times d \times s} \quad (5)$$

Таблица 3 – Точность и память алгоритмов

Алгоритмы	Точность	Память
Static	61%	1
Last decision	56%	1
Bim	68%	8 Кб
Gh	77%	8 Кб+4
Ph	79%	24 Кб
GShare	88%	24 Кб+4
Tage	95%	>100 Кб



В ходе исследования:

- 1) Предложен метод ускорения синтеза предсказателя переходов
- 2) Реализованы алгоритмы, тестовое окружение, проведён сбор статистики
- 3) Проведён анализ алгоритмов и конфигураций, выявлены рекомендации по выбору эффективного предсказателя по характеристикам задачи
- 4) Выведена формула определения целевой точности по параметрам конвейера
- 5) Метод экспериментально подтвержден

Метод позволяет ускорить выбор и проектирование структуры предсказателя переходов на основе параметров конвейера и характеристик решаемых задач.

Использование метода позволит экономить ресурсы при проектировании процессоров, что благоприятно сказывается на общей работоспособности при разработке.

Материалы проекта: <https://github.com/pupyr/SHware/>