

Дипломная работа

Разработка RTL-описания интегрированного микропроцессорного модуля с RISC-архитектурой

Кареев Кирилл Андреевич¹

¹МИЭТ

МИЭТ, 2016

Содержание

1 Мотивация

- Причины создания нового RISC процессора
- Существующие решения

2 Процессор УП-1

- Общее описание
- Набор инструкций

3 Результат

1 Мотивация

- Причины создания нового RISC процессора
- Существующие решения

2 Процессор УП-1

- Общее описание
- Набор инструкций

3 Результат

Чего не хватает другим процессорам

- Простоты работы с машинным кодом и ассемблерным представлением.
- Единой структуры блока исполнения, без сопроцессоров и особых регистровых наборов.
- Открытости RTL-описания.

Чего не хватает другим процессорам

- Простоты работы с машинным кодом и ассемблерным представлением.
- Единой структуры блока исполнения, без сопроцессоров и особых регистровых наборов.
- Открытости RTL-описания.

Чего не хватает другим процессорам

- Простоты работы с машинным кодом и ассемблерным представлением.
- Единой структуры блока исполнения, без сопроцессоров и особых регистровых наборов.
- Открытости RTL-описания.

Что есть в этом процессоре

- Развитый набор команд, позволяющий в большинстве случаев обойтись без макросов и теневых определений.
- Практически все инструкции исполняются за один такт.
- Все инструкции могут использовать все регистры и иметь мгновенные значения для всех входных аргументов.
- Все инструкции относятся к основному набору, т.е. в процессоре нет сопроцессоров.
- Составные блоки конвейера достаточно универсальны для того, чтобы новый набор инструкций можно было ввести сменой лишь одного декодера.
- Открытое RTL-описание - читать и править можно всем :)

Что есть в этом процессоре

- Развитый набор команд, позволяющий в большинстве случаев обойтись без макросов и теневых определений.
- Практически все инструкции исполняются за один такт.
- Все инструкции могут использовать все регистры и иметь мгновенные значения для всех входных аргументов.
- Все инструкции относятся к основному набору, т.е. в процессоре нет сопроцессоров.
- Составные блоки конвейера достаточно универсальны для того, чтобы новый набор инструкций можно было ввести сменой лишь одного декодера.
- Открытое RTL-описание - читать и править можно всем :)

Что есть в этом процессоре

- Развитый набор команд, позволяющий в большинстве случаев обойтись без макросов и теневых определений.
- Практически все инструкции исполняются за один такт.
- Все инструкции могут использовать все регистры и иметь мгновенные значения для всех входных аргументов.
- Все инструкции относятся к основному набору, т.е. в процессоре нет сопроцессоров.
- Составные блоки конвейера достаточно универсальны для того, чтобы новый набор инструкций можно было ввести сменой лишь одного декодера.
- Открытое RTL-описание - читать и править можно всем :)

Что есть в этом процессоре

- Развитый набор команд, позволяющий в большинстве случаев обойтись без макросов и теневых определений.
- Практически все инструкции исполняются за один такт.
- Все инструкции могут использовать все регистры и иметь мгновенные значения для всех входных аргументов.
- Все инструкции относятся к основному набору, т.е. в процессоре нет сопроцессоров.
- Составные блоки конвейера достаточно универсальны для того, чтобы новый набор инструкций можно было ввести сменой лишь одного декодера.
- Открытое RTL-описание - читать и править можно всем :)

Что есть в этом процессоре

- Развитый набор команд, позволяющий в большинстве случаев обойтись без макросов и теневых определений.
- Практически все инструкции исполняются за один такт.
- Все инструкции могут использовать все регистры и иметь мгновенные значения для всех входных аргументов.
- Все инструкции относятся к основному набору, т.е. в процессоре нет сопроцессоров.
- Составные блоки конвейера достаточно универсальны для того, чтобы новый набор инструкций можно было ввести сменой лишь одного декодера.
- Открытое RTL-описание - читать и править можно всем :)

Чего нет в этом процессоре

- Достаточного функционала для реализации какого-либо CISC-набора инструкций.
- Векторных расширений и работы с плавающими числами двойной и четверной точности.
- Упора на малый размер бинарного представления машинного кода.
- Мультиплексирования памяти по частям машинного слова (возможна работа только с целым словом).
- Особых свойств (флаги, префиксы) команд и состояний (ошибка, гипервизор, быстрое прерывание и т.д.) процессора

Чего нет в этом процессоре

- Достаточного функционала для реализации какого-либо CISC-набора инструкций.
- Векторных расширений и работы с плавающими числами двойной и четверной точности.
- Упора на малый размер бинарного представления машинного кода.
- Мультиплексирования памяти по частям машинного слова (возможна работа только с целым словом).
- Особых свойств (флаги, префиксы) команд и состояний (ошибка, гипервизор, быстрое прерывание и т.д.) процессора

Чего нет в этом процессоре

- Достаточного функционала для реализации какого-либо CISC-набора инструкций.
- Векторных расширений и работы с плавающими числами двойной и четверной точности.
- Упора на малый размер бинарного представления машинного кода.
- Мультиплексирования памяти по частям машинного слова (возможна работа только с целым словом).
- Особых свойств (флаги, префиксы) команд и состояний (ошибка, гипервизор, быстрое прерывание и т.д.) процессора

Чего нет в этом процессоре

- Достаточного функционала для реализации какого-либо CISC-набора инструкций.
- Векторных расширений и работы с плавающими числами двойной и четверной точности.
- Упора на малый размер бинарного представления машинного кода.
- Мультиплексирования памяти по частям машинного слова (возможна работа только с целым словом).
- Особых свойств (флаги, префиксы) команд и состояний (ошибка, гипервизор, быстрое прерывание и т.д.) процессора

Чего нет в этом процессоре

- Достаточного функционала для реализации какого-либо CISC-набора инструкций.
- Векторных расширений и работы с плавающими числами двойной и четверной точности.
- Упора на малый размер бинарного представления машинного кода.
- Мультиплексирования памяти по частям машинного слова (возможна работа только с целым словом).
- Особых свойств (флаги, префиксы) команд и состояний (ошибка, гипервизор, быстрое прерывание и т.д.) процессора

1 Мотивация

- Причины создания нового RISC процессора
- Существующие решения

2 Процессор УП-1

- Общее описание
- Набор инструкций

3 Результат

ARM Thumb1

Главные недостатки

- Сложность бинарного представления машинного кода (из-за упора на уменьшенный размер).
- Работа с дробными частями машинного слова.
- Сложность работы с ассемблерным представлением кода (следствие функциональной простоты).

OpenRISC 1000 (mor1kx)

Главные недостатки

- Наличие большого количества состояний процессора.
- Сложность RTL-описания, в основном из-за высокой функциональной развитости.

MIPS32

Главные недостатки

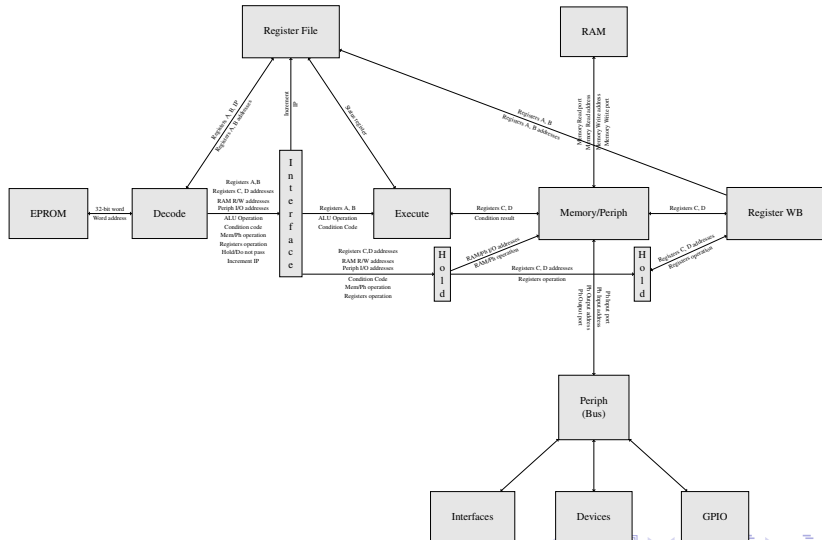
- Относительно сложное построение инструкции
- Большинство реализаций не совместимы друг с другом

- 1 Мотивация
 - Причины создания нового RISC процессора
 - Существующие решения
- 2 Процессор УП-1
 - Общее описание
 - Набор инструкций
- 3 Результат

Характеристики

- 32-битная архитектура
- Набор из 33 (заложено до 128) инструкций
- 32 РОН (Регистра общего назначения) шириной 32 бита с четырёхпортовым интерфейсом (2 чтение, 2 запись + особые линии для PC и LR)
- Регистры PC и LR (счётчик инструкций и адрес возврата) также являются общими (31 и 29 соответственно)
- Однотактовый умножитель с возможностью сохранения всего результата (2 слова)
- Однотактовый комбинированный регистр сдвига (циклический, арифметический и обыкновенный сдвиг)
- Раздельные шины памяти и периферического устройств
- 16 кодов условного исполнения
- Четырёхшаговая архитектура конвейера (Декодирование, Исполнение, Память/Периферия и Регистры)
- Двухпортовое однотактовое ОЗУ ёмкостью 4 КБ (1 Кс) (1- чтение, 1 - запись)

Структурная схема



Структурная схема Устройства

- EPROM - ППЗУ. На выходе 32-битное слово, находящееся по адресу на входе. Текущий размер - 16 кб (4096 слов)
- Register File - регистровый файл. Имеет 32 регистра, 2 порта на чтение (A, B) и 2 порта на запись (C, D) и спецвыходы для IP и Status
- RAM - ОЗУ. Имеет 1 порт на запись и 1 порт на чтение. Текущий размер - 4 кб (1024 слов).
- Periph - Двухнаправленная шина периферических устройств. Имеет два отдельных (адрес+данные) направления - запись и чтение. К ней подключены:
 - GPIO - Ввод-вывод общего назначения - «ноги», с мультиплексорами для выбора дополнительных функций.
 - Interfaces - Различные интерфейсы ввода-вывода (SPI и UART, например)
 - Devices - Различные периферические устройства (RTC и ШИМ-контроллеры, например)

Структурная схема

Стадии конвейера

- Decode - стадия дешифрования входного слова (инструкции) в набор управляющих сигналов. (D)
 - Interface - специальная стадия задержки, нужна для осуществления многотактовых инструкций. (I)
- Execute - стадия исполнения инструкции. Содержит АЛУ и блок вычисления условных кодов. (EX)
- Memory/Periph - стадия комбинированных операций с памятью и шиной периферических устройств. (MEM)
- Register WB - стадия обратной записи в регистры.
- HOLD(n) - специальные подстадии, обеспечивающие удержание управляющих сигналов для синхронизации.

Схема АЛУ

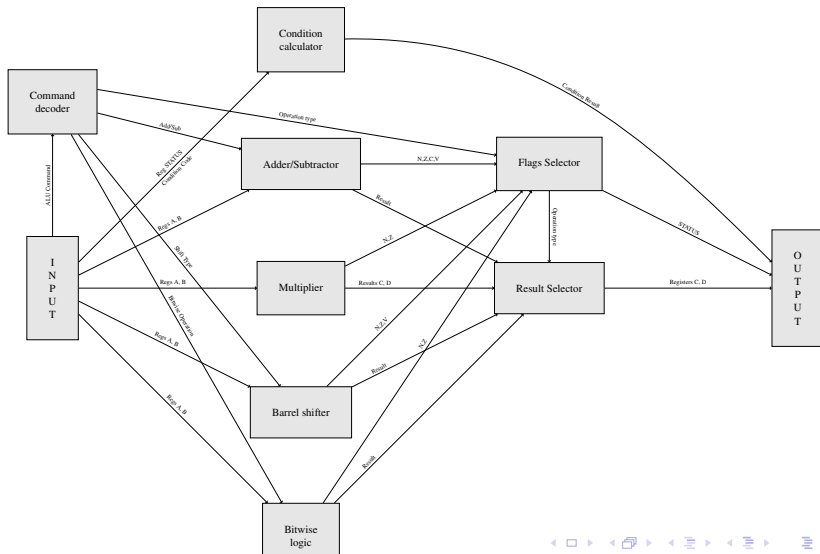


Схема АЛУ (синтез)

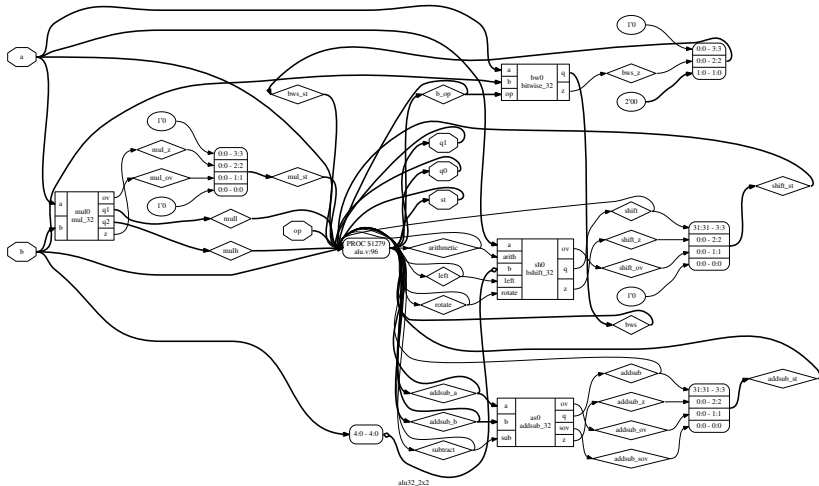


Схема АЛУ

Устройства

- Command Decoder - декодер команды АЛУ во внутренние сигналы управления
- Condition calculator - вычисляет условный результат из регистра STATUS и условного кода
- Adder/Subtractor - комбинированный сумматор-вычитатель $32 \times 32 = 32$
- Multiplier - беззнаковый умножитель $32 \times 32 = 64$
- Barrel shifter - комбинированный сдвиговый регистр $32 \times 5 = 160$
- Bitwise logic - комбинированный блок логических операций, $32 \times 32 = 32$

Периферийные устройства

Адреса

- 0000x - не назначены (защита от случайной перезаписи)
- 0001x - gpio_mux0 (пины 31:0)
- 0010x - gpio_mux1 (пины 63:32)
- 0011x - gpio_mux2 (пины 95:64)
- 0100x - gpio_mux4 (пины 127:96)
- 0101x - gpio_chip1 (gpio_mux0, func0)
- 0110x - gpio_chip2 (gpio_mux1, func0)
- 0111x - gpio_chip3 (gpio_mux2, func0)
- 1000x - gpio_chip4 (gpio_mux3, func0)

Периферийные устройства

Краткое описание

gpio_mux

Мультиплексор ноги устройства на 4 функции на вход и выход. Размер - 32 пина.
На каждый пин отведены два бита, значение которых определяет его функцию.

Регистры:

- xxxx0 - Младшая часть регистра управления
- xxxx1 - Старшая часть регистра управления

gpio_chip

Контроллер GPIO с поддержкой чтения/записи. Размер - 32 в/в.

Состояние каждого в/в управляется соответствующими битами в двух регистрах

Регистры:

- xxxx0 - Регистр VALUE - значение уровня на выходе в/в. 0 - низкий, 1 - высокий
- xxxx1 - Регистр DIR - направление в/в (0 - вход, (значение на выходе - z), 1 - выход)

Флаги (статус)

- (N)egative - отрицательное число, равен 31 биту результата
- (Z)ero - нулевой результат
- (C)arry - беззнаковое переполнение (перенос), например,
 $\text{FFFFFFFFh} + \text{FFFFFFFFh}$ (беззнаковое) = C + FFFFFFFE
- O(V)erflow - знаковое переполнение, например, $7\text{FFFFFFF} + 7\text{FFFFFFF}$ (знаковое) = V - 7FFFFFFE

- 1 Мотивация
 - Причины создания нового RISC процессора
 - Существующие решения
- 2 Процессор УП-1
 - Общее описание
 - Набор инструкций
- 3 Результат

Характеристики

- 6 логических инструкций (or, nor, and, nand, inv, xor).
- 6 видов сдвига (арифметический, логический и циклический), влево и вправо.
- Арифметический операции (сумма, разность, беззнаковое произведение, инкремент/декремент, сравнение)
- Операции потока исполнения (прыжок, вызов и возврат), в т.ч и относительные
- Операции с памятью и стеком (чтение и запись), в т.ч. не прямые
- Операция перемещения регистр-регистр (в т.ч двойная) и пустая операция
- Операции с шиной периферических устройств
- Все операции имеют возможность условного исполнения.

Условные коды

- EQ - равен (первый операнд второму)
- NEQ - не равен
- HE - больше либо равен
- H - больше
- LE - меньше либо равен
- L - меньше
- SHE - знаковый больше либо равен
- SH - знаковый больше
- SLE - знаковый меньше либо равен
- SL - знаковый меньше
- OV - переполнение (знаковое)
- NOV - не переполнение (знаковое)
- POS - положительное число
- NEG - отрицательное число
- AL - всегда (безусловно)

Перечисление инструкций

Логические инструкции

- OR - побитовое ИЛИ двух операндов в третий
- NOR - побитовое ИЛИ-НЕ двух операндов в третий
- AND - побитовое И двух операндов в третий
- NAND - побитовое И-НЕ двух операндов в третий
- INV - побитовая инверсия одного операнда во второй
- XOR - побитовое ИСКЛ. ИЛИ двух операндов в третий
- XNOR - побитовое ИСКЛ. ИЛИ-НЕ двух операндов в третий

Перечисление инструкций

Сдвиговые инструкции

- LSL - логический сдвиг влево
- LSR - логический сдвиг вправо
- ASR - арифметический сдвиг вправо
- ASL - арифметический сдвиг влево
- CSR - циклический сдвиг (вращение) вправо
- CSL - циклический сдвиг (вращение) влево

Перечисление инструкций

Арифметические операции

- ADD - сложение двух операндов в третий
- SUB - вычитание двух операндов в третий
- MUL - беззнаковое перемножение двух операндов в третий(LO) и четвёртый (HI)
- CSG - смена знака операнда во второй
- INC - инкремент операнда во второй
- DEC - декремент операнда во второй
- CMP - сравнение двух операндов (вычитание без сохранения результата)
- CMN - сравнение операнда с инвертированным вторым.
- TST - тест операндов (побитовое И без сохранения результата)

Перечисление инструкций

Переходы

- BR - абсолютный переход по адресу в операнде
- RBR - относительный (малый) переход по адресу в IP+операнд
- BRL - абсолютный переход с запоминанием (в LR)
- RET - возврат (переход по адресу в LR)

Перечисление инструкций

Память и стек

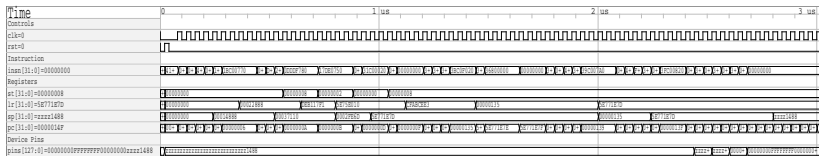
- LDR - загрузка из адреса в операнде
- STR - запись по адресу в операнде
- LDRC - непрямая загрузка из адреса операнд1+операнд2

Перечисление инструкций

Периферия и прочее

- IN - ввод из регистра периферии по адресу в операнде
- OUT - вывод в регистр периферии по адресу в операнде
- MOV - скопировать {операнд1, операнд2} в {операнд3, операнд 4}
- NOP - ничего не делать 1 (или операнд1) такт(ов).

Моделирование



Моделирование

Тестовые воздействия

```
movs 0x14888 -> r30
nop
movs 0x22888 -> lr
add lr, r30 -> r30
add 0x35942, 0xDEADBEAF -> lr
mul lr, r30 -> lr, r30
xor lr, r30 -> r30
csr r30 by 11 -> lr
br to 0x132
[0x132]
out lr to r30
out r30 to lr
brl to r30
str to 16 from r30
mov lr, r30 -> r30, lr
out r30 to lr
movs 0xFFFFFFFF -> r1
out to 0xD from r1
out to 0xF from r1
out to 0x11 from r1
out to 0xE from r1
in from 0xA -> r30
[0x5E771E7D]
brpos to r0
retneg
```

Синтез

Статистика

Был произведён синтез в 3 различные цели:

- Xilinx 7-Series FPGA:
 - Количество ячеек - 16761
 - Количество внутренних путей - 18586
 - Количество внешних путей - 9179
- iCE40 FPGA:
 - Количество ячеек - 74852
 - Количество внутренних путей - 73418
 - Количество внешних путей - 908
- ASIC (библиотека osu025_stdcells):
 - Количество ячеек - 125904 (из них ram - 100673)
 - Количество внутренних путей - 94181 (из них ram - 68842)
 - Количество внешних путей - 10236 (из них ram - 1033)

Результат

- Создано **RTL-описание** процессора УП-1.
- Создана **документация** к процессору УП-1 .
- Произведено **моделирование** процессора УП-1 в симуляторе Icarus Verilog.
- Произведён **синтез** процессора с помощью набора синтезаторов Yosys.