

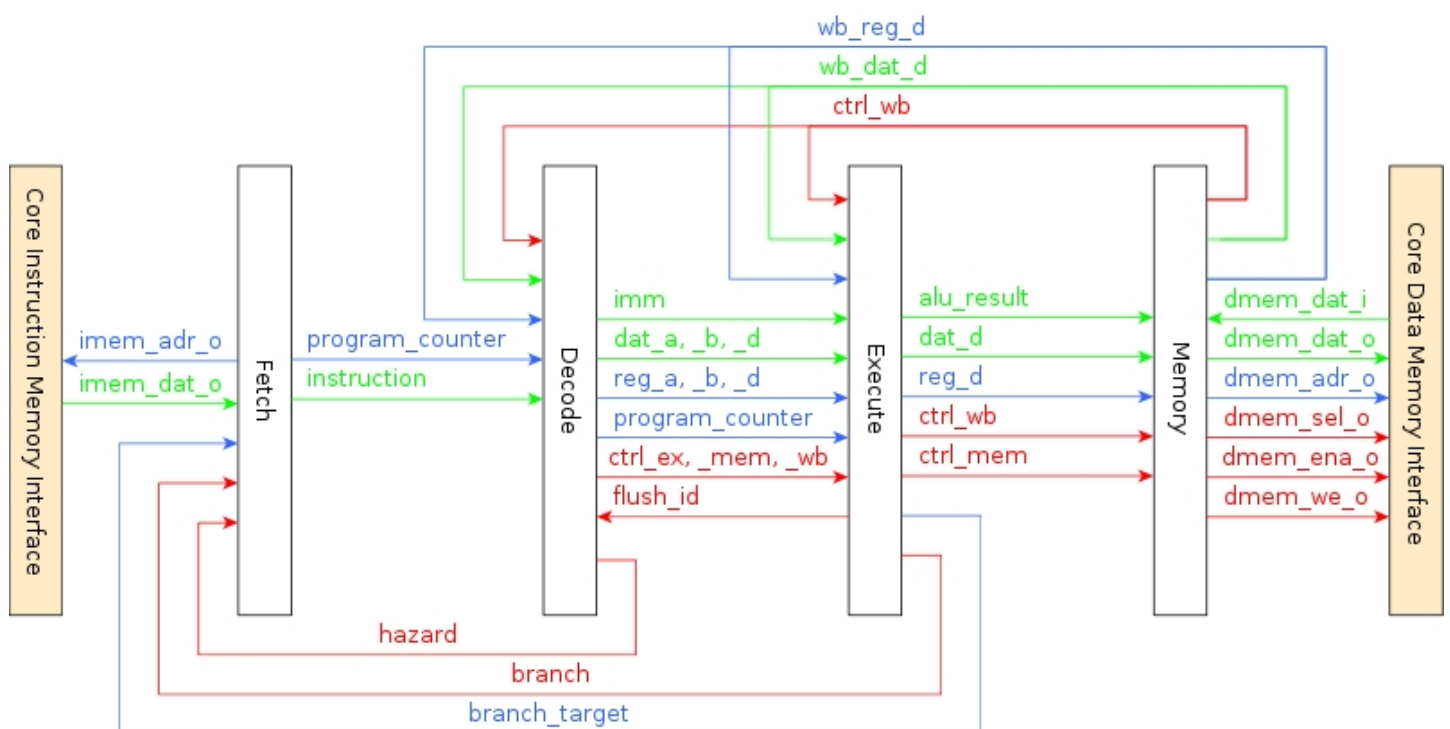
Процессорное ядро MB-Lite

Краткое описание

MB-Lite - компактное 32-разрядное RISC ядро, имеющее набор инструкций совместимый с архитектурой MicroBlaze. Ядро использует 3-стадийный конвейер: получение инструкции из памяти, декодирование инструкции, выполнение и запись результата. Архитектура доступа к памяти - гарвардская, с отдельными шинами инструкций и данных, см. [Шина данных MB-Lite](#). Производительность ядра MB-Lite на стандартном тесте Dhrystone 2 составляет ~1 DMIPS/МГц.

Одно из основных преимуществ ядра - компактность реализации. Ядро написано на языке VHDL и структура кода позволяет конфигурировать RTL-описание в зависимости от задачи. Конфигурация позволяет выбирать различную реализацию регистрового файла, а также включать или отключать синтез опциональных компонентов АЛУ: 32-битного циклического сдвига и (только при синтезе в ПЛИС) 32-битного умножителя.

Структура конвейера



Файлы, необходимые для синтеза

Библиотека work:

core_tb.vhd	Тестовое окружение
core.vhd	Модуль верхнего уровня, объединяющий компоненты ядра
fetch.vhd	Компонент конвейера: получение инструкции из памяти
decode.vhd	Компонент конвейера: декодирование инструкции
execute.vhd	Компонент конвейера: исполнение инструкции
gprf.vhd	Регистровый файл (две возможные архитектуры)
mem.vhd	Модуль управления шинами памяти

Библиотека mblite:

mblite/config_Pkg.vhd	Конфигурационный файл
mblite/core_Pkg.vhd	Пакет с объявлениями компонентов, функций и типов, используемых в ядре
mblite/std_Pkg.vhd	Пакет с объявлениями компонентов, функций и типов, используемых в ядре

Библиотека malt:

malt/malt_cfg.vhd	Конфигурационный файл
malt/malt_defines.vhd	Пакет с объявлениями компонентов, функций и типов, используемых в проекте
malt/malt_pkg.vhd	Пакет с объявлениями компонентов, функций и типов, используемых в проекте
malt/sim_pkg.vhd	Конфигурационный файл

Интерфейсы модуля верхнего уровня core.vhd

Сигнал/шина	Направление	Назначение
clk_i	выход	Тактовый сигнал
rst_i	вход	Сигнал сброса
imem_o	выход	Шина инструкций MB-Lite
dmem_o	выход	Шина инструкций MB-Lite
imem_i	вход	Шина данных MB-Lite
dmem_i	вход	Шина данных MB-Lite
int_i	вход	Сигнал запроса прерывания
int_ack_o	выход	Сигнал подтверждения прерывания

Результаты синтеза на ПЛИС Xilinx Virtex-7

LUT: 946-2200 (в зависимости от конфигурации)
Частота: 200 МГц

Тестовое окружение

Модуль тестбенча core_tb.vhd подключает компонент core, генерирует для него сигнал сброса и частоту 100 МГц и тестирует ядро с помощью простейшей программы из 5 инструкций. Программа производит обращения к фиктивной памяти данных по последовательным адресам. Это проявляется в том, что начиная со времени 135 нс раз в 40 нс сигналы dmem_out.ena_o и dmem_out.we_o принимают состояние 1, а шины dmem_out.adr_o и dmem_out.dat_o инкрементируются на 4 с каждой такой записью начиная с нуля. Состояние шин контролируется процессом и, в случае корректного исполнения, при каждой записи в консоль симулятора с помощью VHDL report будет выводиться сообщение "Core dmem write OK".

Шина данных MB-Lite

MB-Lite является процессором с Гарвардской архитектурой - это означает, что он использует отдельные шины для данных и для инструкций. Обе шины очень просты и совпадают с точностью до отсутствия сигналов предназначенных для управления записью в imem-шине (инструкции только читаются). Типы определяющие эти шины (dmem_in_type и dmem_out_type для данных; imem_in_type и imem_out_type для инструкций соответственно) находятся в файле *mblite/core_pkg.vhd*.

Интерфейс dmem-шины

Сигналы входящие в шину (CFG_DMEM_WIDTH = CFG_DMEM_SIZE = 32, направления шин относительно процессора):

```
type dmem_in_type is record
    -- данные читаемые процессором
    dat_i : std_logic_vector(CFG_DMEM_WIDTH - 1 downto 0);
    -- готовность ('1' - готов, '0' - стоп)
    ena_i : std_logic;
end record;

type dmem_out_type is record
    -- данные записываемые процессором
    dat_o : std_logic_vector(CFG_DMEM_WIDTH - 1 downto 0);
    -- адрес обращения
    adr_o : std_logic_vector(CFG_DMEM_SIZE - 1 downto 0);
    -- маска записи
    sel_o : std_logic_vector(3 downto 0);
    -- '1' = запись, '0' = чтение
    we_o : std_logic;
    -- '1' = обращение в память
    ena_o : std_logic;
end record;
```

Сигнал dmem_in.ena_i предназначен для остановки процессора. Процессор работает только когда dmem_in.ena_i = '1', в противном случае он простаивает. Всякий раз, когда необходимые процессору данные не могут быть получены в такт, когда процессор их ожидает, его необходимо отправлять в простой.

Когда процессор хочет обратиться к данным, он выставляет требуемый адрес на шину dmem_out.adr_o, выставляет направление обращения в dmem_out.we_o и выставляет строб dmem_out.ena_o в '1'. В случае записи (dmem_out.we_o = '1') процессор также выставляет маску записи на шину dmem_out.sel_o (байты соответствующие '1' на sel_o должны быть записаны, соответствующие '0' - нет) и данные для записи на шину dmem_out.dat_o. В случае чтения (dmem_out.we_o = '0') процессор ожидает получить на следующий рабочий такт (когда dmem_in.ena_i = '1') прочитанные данные на шине dmem_in.dat_i. В случае если чтение или запись невозможно произвести за один такт, то процессор необходимо остановить установкой сигнала dmem_in.ena_i в '0' и запустить снова когда данные будут готовы.