

ЛАБОРАТОРНАЯ РАБОТА № 5 "Арифметико-логическое устройство"

1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

1.1. Команды арифметической и логической обработки

Команды арифметической и логической обработки обеспечивают арифметическую и логическую обработку информации в различных формах ее представления.

Для каждой формы представления чисел в архитектуре системы команд предусматривается определённый стандартный набор операций.

Помимо вычисления результата, формируются признаки (флаги) в АЛУ, характеризующие результат (например, Zero, Negative, Overflow, Carry).

Стандартный набор операций с целыми числами:

- двухместные (с двумя операндами) арифметические операции: «+», «-», «*», «/».

- одноместные арифметические операции. Например, абсолютное значение, изменение знака операнда.

- операции сравнения, устанавливающие признаки, характеризующие соотношения между величинами (=, <, >, <=, >=).

Часто дополняют - вычисление остатка от целочисленного деления, сложение с учётом переноса, вычитание с учётом заема, инкремент, декремент.

Типичный набор операций с числами в форме с плавающей запятой (ПЗ):

- основные арифметические операции: «+», «-», «*», «/».

- операции сравнения, устанавливающие признаки, характеризующие соотношения между величинами (=, <, >, <=, >=).

- операции преобразования: формы представления (между формой с фиксированной запятой (ФЗ) и плавающей запятой), формата представления (с одинарной и двойной точностью).

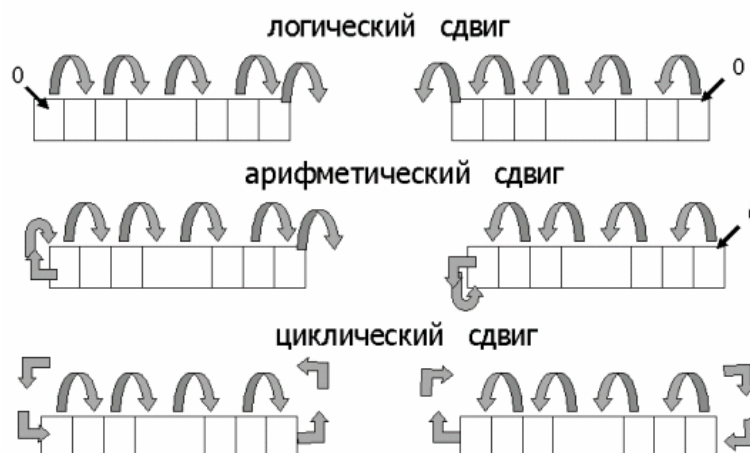


Рис. 1.1. Типы сдвиговых операций

Логические команды предназначены для выполнения логических операций, как над отдельными битами слов, так и целыми байтами, словами и т.п.

Минимальный набор - «НЕ», «И», «ИЛИ», «сложение по модулю 2»

В дополнение практически во всех архитектурах системы команд (АСК) предусмотрены команды логического, арифметического и циклического сдвигов (рис 1.1).

Логический сдвиг – это сдвиг, при котором уходящий бит уходит, не влияя на оставшиеся биты, а на место появившегося бита записывается бит 0. Пример работы операции сдвига: число 10101010_b дает число 01010100_b, если сделать сдвиг влево на 1 бит, и число 01010101_b, если сделать сдвиг вправо на 1 бит.

Арифметический сдвиг. При этом сдвиге слово рассматривается не просто как группа битов, а как целое число в дополнительном коде. При сдвиге влево ведет себя как логический сдвиг, при сдвиге вправо: уходящий бит уходит, не влияя на оставшиеся биты, а на место появившегося бита устанавливается бит, соответствующий знаку. Пример работы операции сдвига: число 11111010_b=-6 дает число 11110100_b=-12, если сделать сдвиг влево на 1 бит, и число 11111101_b=-3, если сделать сдвиг вправо на 1 бит.

Легко заметить, что при арифметическом сдвиге сдвиг влево соответствует умножению на 2, а сдвиг вправо делению на 2.

Циклический сдвиг. При этом сдвиге уходящий бит появляется на месте появившегося. Пример работы операции сдвига: число 11111010_b дает число 11110101_b, если сделать сдвиг влево на 1 бит, и число 01111101_b, если сделать сдвиг вправо на 1 бит.

Операции с десятичными числами:

Ранее широко использовались специальные форматы хранения - двоично-десятичный код (BCD - Binary Coded Decimal).

Два формата: зонный и упакованный. Умножение и деление возможно только в зонном формате.

В ВМ первых поколений в АСК были специальные команды («+», «-», «*», «/») для десятичных чисел.

В АСК современных машин, соответствующие вычисления имитируются с помощью команд целочисленной арифметики.

1.2. Разработка АЛУ

Для разработки блока АЛУ рекомендуется следующая последовательность действий:

- 1) Разработать непосредственно блоки, выполняющие каждую из заданных по варианту операций;
- 2) Реализовать выбор операции в АЛУ в зависимости от управляющих сигналов (на формирование которых влияет дешифрованная команда), причем при реализации функций использовать только логические операции;
- 3) Определить источники операндов, а также приемники результата;
- 4) Разработать блок управления АЛУ;

5) Проверить работу АЛУ как отдельного модуля, сформировав управляющие сигналы самостоятельно;

6) Проверить работу всего проекта, предварительно задав в памяти команд – команды для работы с АЛУ.

В качестве примера приведено АЛУ, реализованное на одноадресном регистровом ЗУ и комбинационной логике.

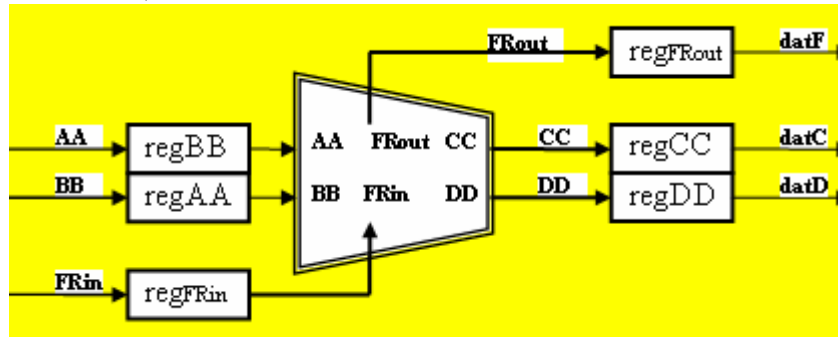


Рис. 1.2. АЛУ на одноадресном ЗУ и комбинационной логике

В одноадресном ЗУ невозможно одновременное обращение по двум адресам и считывание/запись двух данных. Для временного хранения двух операндов в схеме установлены буферные регистры **regAA** и **regBB**, в которые через мультиплексор поочередно загружаются операнды А и В с шины данных, куда их “выбрасывает” АЛУ-команда. Некоторые АЛУ-команды используют признаки-флаги предыдущих команд и/или вырабатывают свои признаки (флаги) результатов: выходные флаги заносятся в **regFRout**, входные флаги поступают из **regFRin**, куда дублируются из регистра флагов **FR** по шине данных. Через некоторое время на выходе появляются АЛУ-результаты: **CC**, **DD** и **FRout**, которые фиксируются в выходных регистрах **regCC**, **regDD** и **regFRout**. Отсюда по шине данных они переносятся в блок РОНов: **regCC** =>**RON#0**, **regDD** =>**RON#1** – соответственно и **regFRou** =>**regFR** блока регистра флагов УУ. Последовательность управляющих сигналов, реализующих перечисленные действия для команд NOT, AND генерируется блоком управления АЛУ в главном блоке управления.

2. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

При выполнении лабораторной работы необходимо использовать наработки из предыдущей лабораторной работы.

К архитектуре системы из предыдущей лабораторной работы добавить блок АЛУ и управление для него.

Необходимо выполнить полный цикл заданных по варианту команд АЛУ. Вспомним, что цикл команды состоит из следующих фаз:

- выборки команды и формирования адреса следующей команды;
- декодирования команды;
- формирования исполнительных адресов операндов;
- выборки операндов;
- исполнения операции;
- записи результата.

Ненужные для заданной команды фазы отбрасываются. В процессе выполнения фаз должны по назначению использоваться специальные регистры (IP, IR, AR, DR, FR).

Арифметические команды необходимо выполнить на логических элементах.

Семейство ПЛИС для реализации - Flex10K (изменить семейство ПЛИС(Family) можно в настройках проекта в ветке Device или в пункте меню Assignments->Device).

ВАРИАНТЫ ЗАДАНИЙ К ЛАБОРАТОРНОЙ РАБОТЕ № 4

№	Команда сдвига	Арифметическая команда, адресация операндов	Логическая команда
1	SLL(Shift Left Logical)	INCS, непосредственная	NOT
2	SRL(Shift Right Logical)	ADDC, прямая регистровая	AND
3	SLA(Shift Right Arithm.)	SUB, косвенная регистровая	OR
4	SRA(Shift Right Arithm.)	CMR, прямая регистровая	NOR
5	ROL(Rotate Left Log.)	INCS, прямая	NAND
6	ROL(Rotate Right Log.)	ADDC, косвенная регистровая	NOTZ
7	SLA(Shift Right Arithm.)	SUB, прямая регистровая	XOR
8	SRA(Shift Right Arithm.)	CMR, прямая	NXOR

СОДЕРЖАНИЕ ОТЧЕТА ПО ЛАБОРАТОРНОЙ РАБОТЕ №4

1. Задание.
 - 2.1. Схема блока, самого верхнего в иерархии проекта.
 - 2.2. Схемы блока АЛУ и входящих в него блоков
 - 2.3. Схема блока управления АЛУ
3. Результаты моделирования.