**Построение собственной архитектуры ЭВМ.**

Книга от среднестатистического ученика 10 класса.

**В гостях у “сказки”**

**о науке и логике**

**через испытания к**

**“великим” знаниям**

**Содержание**

|  |  |
| --- | --- |
| * Свойства проекта | 3 |
| * Вступление | 4 |
| * Теоретическая часть | 5 |
| * + Логические вентили | 6 |
| * + Архитектуры | 8 |
| * + Структура ЭВМ |  |
| * + - Структура Процессора |  |
| * + - * АЛУ |  |
| * + - * + Единая структура |  |
| * + - * + Блочная структура |  |
| Сумматор |  |
| Блок логических действий |  |
| Блок сравнения |  |
| Устройство сдвига |  |
| Дополнительные блоки |  |
| * + - * УУ |  |
| * + - * + Устройство декодирования команд |  |
| * + - * + Устройство тактирования |  |
|  |  |
| * Практическая часть |  |

**Свойства проекта**

**Цель:**

|  |
| --- |
| * Собрать собственную архитектуру процессора. (продукт проекта 1) |
| * Рассказать о логических схемах в формате книги. (продукт проекта 2) |

**Задачи:**

|  |
| --- |
| * Найти информацию об архитектурах. |
| * Найти похожие проекты и рассказать о них. |
| * Рассказать факты, касающиеся данной темы, для большего понимания. |
| * Собрать информацию о структуре компонентов и их разновидностях в ЭВМ. |
| * Предложить собственные идеи о структуре компонентов и их разновидностях в ЭВМ. |
|  |
| * Составить набор выполняемых команд собственной архитектуры. |
| * Составить логическую схему собственной архитектуры. |

**Актуальность:**

|  |
| --- |
| * Малое количество простых индивидуальных архитектур, как процессоров, так и ЭВМ. |
| * Отсутствие свободной для понимания информации в Интернете. |

**Вступление**

Здравствуйте, данная книга является продуктом (и собственно копией) индивидуального проекта учащегося 10 класса. Тема, которую я затрагиваю очень редка, и мало кому интересна. Наука изучающая Логические схемы забывается среди населения из-за перехода в промышленность. Крупные компании монополизируют рынок микросхем, а малые производства, не имея многих знаний в данной сфере, практически не способны развиваться. Самые основные устройства и их схемы в интернете, конечно, доступны, но разнообразия почти что нет. Это я имел в виду лишь современную информацию, а литература советского времени довольно богата знаниями. Что говорит об остановке развития науки среди доступного населению пространства. Основные исследования проводят крупные компании, они патентуют почти всё, ведь в ином случае произойдёт потеря прибыли. На момент 2020г. компании ведут яростную борьбу за потребителя, так будет всегда, поэтому в доходах они особо заинтересованы. По данным причинам энтузиасты, желающие сделать что-то и зачем-то, не могут проникнуть в данную сферу. А ведь таких людей не мало, и я сам знаю 6 человек (правда они меня не знают). Так что я вовлечённый в данную научную сферу решил написать данный проект.

Проект рассказывает о создании собственной простейшей архитектуры ЭВМ. Современные ЭВМ как правило имеют процессор, то есть я не буду рассказывать о создании просто процессора, я решил замахнуться и спроектировать ЭВМ целиком. Как и вам полезно, можете прочитать об интересующем вас элементе, так и мне есть чем заняться.

**Теоретическая часть**

Перед началом собственно проектирования нужно понимать, что мы собрались проектировать.

ЭВМ – Электронно-Вычислительная Машина, выполняет вычисления используя электроэнергию. Вы ведь помните, что такое ламповый компьютер, первейший класс ЭВМ? Тогда и названия “компьютер” не было. Они выполняли различные логические и математические задачи. Потребляли много энергии и были размером в 1-4 комнат. А потом с изобретения Транзистора началась новая эра.

Транзистор был отличной заменой Вакуумным электронным лампам. Потребления энергии меньше, размеры намного меньше, способ эксплуатации легче и надёжность выше. Потом люди додумались делать несколько транзисторов в одном корпусе, придумали литографию и начали выпускать первые микросхемы, а там и до процессоров не далеко было.

И да! Процессор как термин появился позже ЭВМ. Но у многих ЭВМ структура позволяла наличие Процессора, как главного вычислительного блока.

Теперь по существу…

Что нам надо знать для проектирования собственной ЭВМ? На самом деле, очень много. Начиная с логических вентилей, заканчивая устройством адресации доп.устройств.

**Логические вентили**

Логические вентили (Логические элементы ЛЭ) - базовый элемент цифровой схемы, выполняющий элементарную логическую операцию, преобразуя таким образом множество входных логических сигналов в выходной логический сигнал.

А проще говоря это, элемент выполняющий какую-то простейшую логику основываясь на входящих сигналах. Они могут иметь как 1 вход, так и 8, или более. И могут состоять из Электронных ламп, транзисторов или чего-нибудь ещё что будет сказываться на их характеристиках. Но всегда имеют 1 выход. Для двоичной логики Логических вентилей существует ровно 7.

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Обозначение | | Описание |
| **Международное** | **IEC 60617-12 : 1997** |
| NOT  (НЕ) |  |  | Инвертирует входящий сигнал. |
| AND  (И) |  |  | Выдаёт сигнал только при наличии сигнала на всех своих входах. |
| OR  (ИЛИ) |  |  | Выдаёт сигнал только при наличии сигнала на любом из своих входов. |
| XOR  (Исключающее ИЛИ) |  |  | Выдаёт сигнал только при наличии сигнала на одном любом из своих входов. |
| NAND  (И-НЕ) |  |  | Выдаёт сигнал только при отсутствии сигнала на хотя бы одном из всех своих входов. |
| NOR  (ИЛИ-НЕ) |  |  | Выдаёт сигнал только при отсутствии сигнала на всех своих входах. |
| XNOR  (Исключающее ИЛИ-НЕ) |  |  | Не выдаёт сигнал только при наличии сигнала на одном любом из своих входов. |

|  |  |
| --- | --- |
| Название | Таблица истинности |
| NOT  (НЕ) | |  |  |  |  | | --- | --- | --- | --- | |  |  |  |  | | A |  | **B** | **Out** | | 0 |  | - | 1 | | 1 |  | - | 0 | | - |  | - | - | | - |  | - | - | |  |  |  |  | |
| AND  (И) | |  |  |  | | --- | --- | --- | |  |  |  | | A | **B** | **Out** | | 0 | 0 | 0 | | 1 | 0 | 0 | | 0 | 1 | 0 | | 1 | 1 | 1 | |  |  |  | |
| OR  (ИЛИ) | |  |  |  | | --- | --- | --- | |  |  |  | | A | **B** | **Out** | | 0 | 0 | 0 | | 1 | 0 | 1 | | 0 | 1 | 1 | | 1 | 1 | 1 | |  |  |  | |
| XOR  (Исключающее ИЛИ) | |  |  |  | | --- | --- | --- | |  |  |  | | A | **B** | **Out** | | 0 | 0 | 0 | | 1 | 0 | 1 | | 0 | 1 | 1 | | 1 | 1 | 0 | |  |  |  | |
| NAND  (И-НЕ) | |  |  |  | | --- | --- | --- | |  |  |  | | A | **B** | **Out** | | 0 | 0 | 1 | | 1 | 0 | 1 | | 0 | 1 | 1 | | 1 | 1 | 0 | |  |  |  | |
| NOR  (ИЛИ-НЕ) | |  |  |  | | --- | --- | --- | |  |  |  | | A | **B** | **Out** | | 0 | 0 | 1 | | 1 | 0 | 0 | | 0 | 1 | 0 | | 1 | 1 | 0 | |  |  |  | |
| XNOR  (Исключающее ИЛИ-НЕ) | |  |  |  | | --- | --- | --- | |  |  |  | | A | **B** | **Out** | | 0 | 0 | 1 | | 1 | 0 | 0 | | 0 | 1 | 0 | | 1 | 1 | 1 | |  |  |  | |

Логику работы этих элементов можно записать в таблицу, которая называется Таблица истинности. Такие таблицы могут быть бесконечны! Для в том, что они дают понять логику работы и возможность взглянуть на схему, с другой стороны.

Также стоит запомнить и научиться составлять логические формулы. Чем-то они похожи на удлинённые физические или математические формулы, но здесь используются определённые символы (которые вам тоже стоит понимать). С помощью этих формул можно посмотреть на логику работы с математической стороны, что может помочь с оптимизацией схемы.

|  |  |  |
| --- | --- | --- |
| Название | Обозначение | Описание |
| Инверсия |  | Инвертирует значение.  Соответствует вентилю НЕ. |
| Конъюнкция | , | Умножает значения.  Соответствует вентилю И. |
| Дизъюнкция | , | Складывает значения.  Соответствует вентилю ИЛИ. |
| Эквиваленция | , | Сравнивает значения.  Соответствует вентилю Исключающее ИЛИ-НЕ. |
| Импликация |  | Если A то возвращается B, иначе 1.  Соответствует вентилю И. |

При оптимизации необходимо учитывать радиоэлектронную схему логических вентилей, их структуру. Например, Элемент И сложнее чем элемент И-НЕ, разница всего на 1-3 Транзистора вам принесёт много проблем с реализацией вашей схемы. Конечно, это зависит от технологии Логического вентиля (ТТЛ, МОП, КМОП, итд). Но в любом случае рекомендую оптимизировать свои схемы.

**Архитектуры**

Архитектура, это довольно общее понятие, собирающее в себе много определённых терминов. Она описывает как структуру с функциями, так и характеристики. Легче будет усвоить информацию разбирая отдельные классы Архитектур, которые соответствуют нашей теме.

Начнём с Архитектуры ЭВМ. Коих имеется много, но основными являются всего два. Это Архитектура Фон-Неймана и Архитектура Гарварда.

|  |  |  |
| --- | --- | --- |
| Фон Нейман |  | Гарвард |
| |  | | --- | | Общая  память | |  | | Процессор | |  | | Прочие  устройства | | |  |  |  | | --- | --- | --- | | Память  данных |  | Память  команд | |  |  |  | | Процессор | | | |  | | | | Прочие  устройства | | | |

Обе этих архитектур совершенно различны и имеют разные задачи. Например, Архитектура Фон-Неймана позволяет написать и запустить программу на уровне машинного кода (Ассемблер), что мы и встречаем в современных компьютерах, а Архитектура Гарварда используется в устройствах где всегда будет выполнятся одна и та же программа, как в микроконтроллерах. Так же стоит отметить, что Архитектура Фон-Неймана уступает по производительности Архитектуре Гарварда, из-за ограничений на запись и чтение данных. Прирост составляет 20% при использовании одноканальной памяти.

Архитектура Процессора — это совокупность главных принципов его конструирования, общая схема расположения деталей и схема взаимодействия программного обеспечения с чипом. И данных Архитектур огромное множество, что были созданы различные классификации. Такие как: RISC, TTA, OISC, CISC, итд. А уже, например, Zen3, M68k, x86 и Ice Lake это непосредственно Архитектуры, которые мы детально рассмотреть увы не сможем. Единственный выход — это разбор классов архитектур, которые недавно упоминались.