общего понимания организации ЭВМ.

Непосредственное проектирование VHDL модели выполнялось модульно. В структуру MC6800 входят: АЛУ, 2 8-разрядых аккумулятора, 2 16-разрядных индексных регистра, программный счетчик, регистр команд и устройство управления. Все модули, кроме УУ, описывались по стандартным методам и использовались шаблоны, встроенные в Xilinx ISE.

В конечном счете архитектура MC6800 следующая. Общая шина данных представляет собой 3 8-разрядные шины. Шина A и B связывает выходы регистров с информационными входами АЛУ соответственно, а шина C – выход АЛУ с входами регистров. Передача данных между шинами A(B) и C осуществляется через АЛУ. Так же шины B и C выходят на двунаправленный буфер данных.

Самый сложный компонент – устройство управления. Устройство представляет из себя конечный автомат, состояниями которого является этапы выполнения команд процессора. Вторая проблема для начинавших – разобраться в методах кодирования команд процессора. Как правило, каждая команда – набор элементарных действий, которые повторяются в разной последовательности и в разном количестве от команды к команде. Эти элементарные действия и их последовательность кодируются в команду. Понимание принципов кодирования помогает описать устройство управление очень лаконично, не расписывая каждую команду в отдельности. Реализация устройства управления MC6800 и была основана на конечном автомате, который реализовывал разные стадии выполнения программы. Конечный автомат определяет стадию выполнения команды и в заданные моменты времени подает сигналы управления на все модули (пример: записи данных в аккумулятор, нужную команду АЛУ). По окончанию выполнения команды - переходит на состояние выборки следующей команды.

Только к концу проектирования, были выявлены несколько методов оптимизации кода на основе кодирования команд, описанной выше.

В свете выше сказанного, хочется отметить что проектирование любого цифрового устройства является решением комплексной задачи аналитики и проектирования с применяем лучших решений в данной области. А применение разнообразных САПР и HDL позволяет оптимизировать работу проектирования, обеспечивает ее гибкость. Данная работа является прямым применением всех наших знаний, полученных за время обучения в нашем ВУЗе.

Список использованных источников:

1. Perry D.L. VHDL: Programming by Example. Fourth Edition. McCraw-Hill, 2002. – 476 p.
2. Бибило П.Н. Основы языка VHDL. Второе издание. — М.: Солон-Р, 2002. — 224 с

РАЗРАБОТКА ИГРОВОГО ПРОЕКТА НА UNREALENGINE 4

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Корсаков А. А.

Разработка игр является комплексной и весьма сложной задачей. Основная сложность заключается в том, что игровые проекты включают в себя множество областей: графика, анимация, обработка огромных объёмов данных в реальном времени, искусственный интеллект, геймдизайн, социальное программирование, повествование, взаимодействие с игроками и так далее. Над проектом, обычно, трудятся команды из огромного числа специалистов в совершенно разных областях.

Можно утверждать, что на данный момент геймдев как никогда близок к рядовым пользователям. С развитием индустрии начала появляться профильная литература, курсы и специальности в учебных заведениях, а пользователи всё чаще пытаются делать свои проекты, тем самым продвигая такое направление, как Инди-разработка.

Я и моя команда не стала исключением. Нам пришлось пройти долгий путь, впитывая в себя огромные объёмы информации, перенимая опыт успешных коллег из-за рубежа и, попутно, экспериментировать и выносить для себя собственные гипотезы и умозаключения.

Для своей работы мы выбрали игровой движок UnrealEngine 4, который, если можно так выразиться, является своеобразным САПРом для создание игр. Его преимуществами является бесплатная модель распространения и, что самое важное, открытый код, который позволяет переписывать любые аспекты системы под свои нужды. UE4 базируется на C++, что даёт ему ощутимый выигрыш в скорости работы относительно конкурентов.

Основными проблемами при разработке игровых проектов являются обеспечение наибольшего быстродействия, организация архитектуры таким образом, чтобы возможно было в кратчайшие сроки вносить изменения и расширять систему, необходимость создания большого количества сопутствующих продуктов, призванных повысить качество и скорость разработки.

Развитие игровой индустрии тесно связано с общим прогрессом в области IT. В частности, на данный

момент именно геймдев сильнее всего стимулирует развитие технологий, связанных с компьютерной графикой и рендером. Именно в разрезе игр впервые обрело законченный вид такое направление, как «виртуальная реальность». Так же, растущие объёмы обработки подталкивают к созданию всё более быстродейственных алгоритмов.

Как уже отмечалось ранее, любой контент, который создаётся для игрового проекта, должен обладать максимальной гибкостью, иметь возможность быстро изменять параметры и расширяться. Как пример можно привести создание искусственного интеллекта. Ограничения по производительности сразу ставят жесткие рамки для написания логики. Так же, крайне важно, чтобы, написав один модуль, можно было получить множество совершенно различных ИИ, регулируя лишь несколько параметров. Благодаря этому, экономится не только время на разработку, но и уменьшается вероятность возникновения ошибок.

Так же, при разработке игрового проекта, большое значение имеет раннее прототипирование. Крайне важно иметь возможность быстро проверить идею на жизнеспособность, затратив на это минимум сил и времени. Данный этап во многом зависит от технологий, которые используются при разработке, и от навыков команды. На данном этапе крайне важно понимать, что качество не имеет никакого значения. Основная идея – посмотреть на то, как будет воспринят итоговый продукт потенциальной аудиторией.

Подводя итог: разработка игр является комплексной задачей, которая за своей кажущейся несерьёзностью скрывает внушительный объем решаемых технических задач и, в общем, вовсе не является тривиальной. Популярность данного направления в данный момент привлекает всё более серьёзные вложения и, как следствие, технологии в геймдев и превращает его в серьёзного игрока на рынке IT.

Список использованных источников:

1. Документация к UnrealEngine 4,URL: https://docs.unrealengine.com/latest/INT/
2. ScottRogers "Levelup! TheGuidetoGreatVideoGameDesign". – Wiley, 2014. – 492 с.

СИСТЕМАУПРАВЛЕНИЯ КОДОВЫМ ЗАМКОМ НА БАЗЕ ПЛАТФОРМЫ ARDUINOMEGA 2560

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Порхун М. И.

Качинский М. В. – кандидат технических наук, доцент

В современном мире одной из главных задач является обеспечение безопасности. Это обязывает использовать системы контроля доступа к информации. Одним из представителей такой системы является кодовый замок.

Применение подобного способа контроля доступа в помещение подразумевает использование секретной комбинации десятичных цифр (в нашем случае четырёх), позволяющей осуществлять управление кодовым замком. Для реализации системы был выбран микроконтроллер ATmega 2560, как одна из самых популярных платформ для реализации различных устройств. Платформа запрограммирована в соответствии с алгоритмом работы устройства на языке, схожим с C/C++. Для проверки работы устройства выбрана система автоматизированного проектирования Proteus.

Данный кодовый замок обеспечивает:

1) Доступ в помещение только после набора секретной кодовой комбинации, представляющей собой четырёхзначное десятичное число;
2) Предоставление ограниченного количества попыток (в нашем случае трёх) набора секретной кодовой комбинации;
3) Звуковуюи визуальную сигнализацию, а также блокировку двери на пять минут при попытке несанкционированного доступа;
4) Оперативную смену секретной кодовой комбинации;
5) Подсветкуклавиатуры при приближении человека к двери;
6) Оригинальное звуковое сопровождение при открытии двери;

Структурная схема устройства приведена на рисунке 1.