Приложения

Приложение А. Пример синтеза устройства управления ЦП

А.1. Пусть необходимо синтезировать УУ ЦП, состав, коды и описания команд машинного языка которого представлены в табл. А.1. Предполагается, что операнды и коды команд являются 8-битовыми, адреса команд – 6-битовыми.

Примечание. Естественно, ЦП с таким набором команд и с такой разрядностью адреса не имеет практического смысла, однако он удобен для пояснения принципов синтеза и реализации УУ.

Таблица А.1

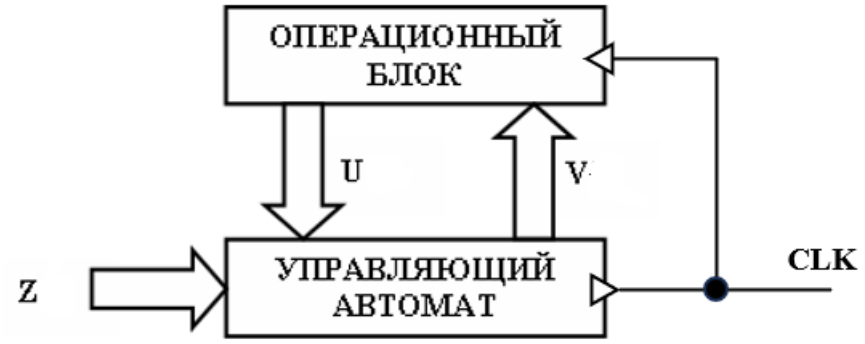
Система команд, подлежащая реализации синтезируемым УУ

Команда	**Двоичный код команды**	**Описание команды**
ADD A, B	00xxxxxx (x – состояние бита безразлично)	Сложение содержимого регистра-аккумулятора с числом *B* на внешних выводах ЦП и запись результата в аккумулятор
SUB A, B	01xxxxxx (x – состояние бита безразлично)	Вычитание числа *B* на внешних выводах ЦП из содержимого регистра-аккумулятора и запись результата в аккумулятор
JZ ADDR	10AAAAAA (AAAAAA – адрес перехода)	При нулевом содержимом аккумулятора – переход по адресу, содержащемуся в 6-и младших битах кода команды. При ненулевом содержимом аккумулятора – переход к следующей команде
JMP ADDR	11AAAAAA (AAAAAA – адрес перехода)	Безусловный переход по адресу, содержащемуся в 6-и младших битах кода команды

А.2. Синтез УУ будем осуществлять на основе классической теории цифровых автоматов. В соответствии с ней, любое цифровое устройство программно-управляемой обработки информации строится по структурной схеме, приведенной на рис. А.1 [А.1]. Оно включает в себя:

- управляющий автомат; применительно к ЦП МК (см. рис. 2.1) – устройство управления;

- операционный блок (ОБ), т. е. совокупность исполнительных функциональных узлов и блоков; в ЦП к ним относятся АЛУ и СОЗУ.



Z – код операции
U – код состояния операционного блока
V – код, управляющий операционным блоком
CLK – сигнал синхронизации

Рис. А.1. Обобщенная структурная схема цифрового устройства программно-управляемой обработки информации

Входными сигналами управляющего автомата служат:

- код операции *Z*, поступающий из памяти;

- код *U* состояния ОБ (называемый также оповещающими сигналами), поступающий с ОБ и используемый при выполнении команд условных переходов.

Управляющий автомат, в соответствии с кодом команды и, в общем случае, с кодом состояния ОБ, вырабатывает код *V* управления ОБ.

Тактирование как управляющего автомата, так и операционного блока осуществляется единым сигналом синхронизации. Во избежание так называемых «гонок» срабатывание функциональных узлов операционного блока и управляющего автомата должно осуществляться по различным фронтам синхросигнала; например, по переднему и по заднему соответственно (см. рис. А.1).

Синтез УУ ЦП с приведенной в табл. А.1 системой команд будем осуществлять на базе представления ЦП как устройства, реализуемого по обобщенной структурной схеме, приведенной на рис. А.1.

А.3. Определяем состав и назначение входов синтезируемого УУ. Как следует из табл. А.1:

- в качестве кода операции (*Z*) синтезируемого УУ служат 2 старших бита кода команды, состояния которых обозначим *z0* и *z1*;

- для реализации системы команд, приведенной в табл. А.1, требуется код состояния ОБ (*U*), содержащий только один бит – признак нулевого содержимого аккумулятора, состояние которого обозначим *u0*.

А.4. Определяем состав и назначение выходов синтезируемого УУ, т. е. битов кода управления ОБ (*V*), для чего, в первую очередь, необходимо определить состав ОБ.

В соответствии с системой команд, подлежащей реализации синтезируемым УУ (см. табл. А.1), а также с обобщенной структурной схемой ЦП, приведенной на рис. 2.1, ОБ должен содержать:

- комбинационное 8-битовое АЛУ, выполняющее две арифметические операции – сложения и вычитания;

- 8-битовый регистр-аккумулятор с параллельной загрузкой данных;

- схему формирования признака нулевого содержимого аккумулятора (единственного сигнала состояния ОБ, требующегося при реализуемой системе команд);

- 6-битовый программный счетчик с возможностью работы в 2-х режимах – прямого счета и параллельной загрузки (при выполнении команд переходов);

- 8-битовый регистр команды (точнее – кода команды) с параллельной загрузкой данных, который на «классической» структурной схеме ЦП (см. рис. 2.1) включается в состав ЦП, но с точки зрения теории цифровых автоматов должен входить в состав ОБ, т. к. является устройством, управляемым УУ.

На основании вышесказанного, назначаем следующий состав битов кода, управляющего ОБ:

- бит *v0*, управляющий режимом работы АЛУ; пусть при его нулевом состоянии выполняется операция суммирования, а при единичном – вычитания;

- бит *v1* разрешения (при его единичном состоянии) или запрета (при нулевом) записи данных в аккумулятор;

- бит *v2* разрешения (при его единичном состоянии) или запрета (при нулевом) загрузки кода в регистр команды;

- бит *v3* разрешения (при его единичном состоянии) или запрета (при нулевом) подачи тактового импульса на программный счетчик;

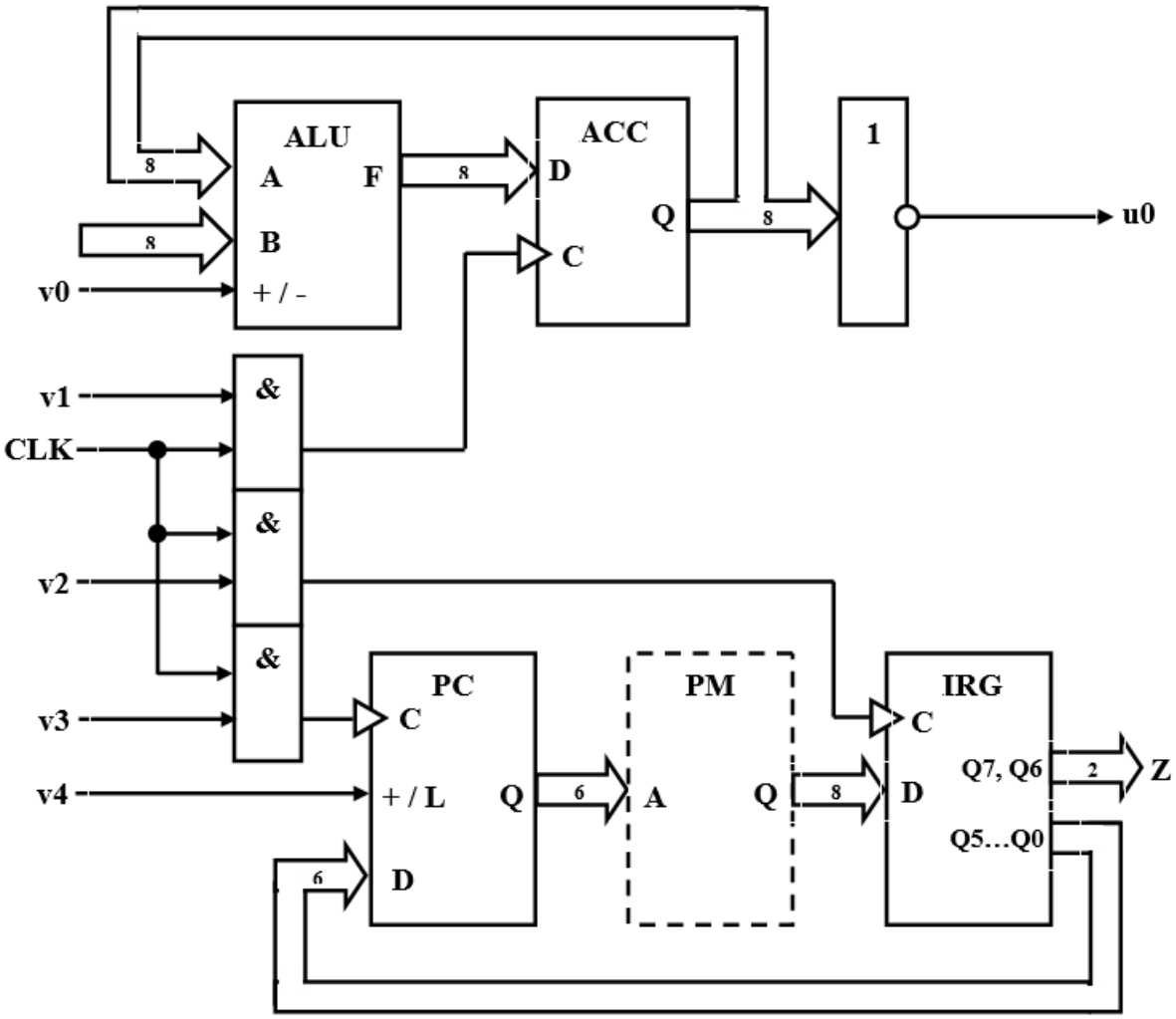
- бит *v4*, управляющий режимом работы программного счетчика; пусть при его нулевом состоянии счетчик работает в режиме суммирования, а при единичном – в режиме параллельной загрузки адреса.

При этом ОБ строится по функциональной схеме, приведенной на рис. А.2. На нем, кроме узлов и блоков собственно ОБ, также показана память программ, для демонстрации того, куда подаются сигналы с выходов программного счетчика и откуда поступают коды команд.

Разрешение / запрет загрузки аккумулятора и регистра команды, а также подачи тактовых импульсов на программный счетчик осуществляется посредством логических элементов 2И, управляемых битами *v1, v2* и *v3* соответственно. Формирование сигнала – признака нулевого содержимого аккумулятора реализуется логическим элементом 8ИЛИ-НЕ.

В рассматриваемом примере все регистры ОБ срабатывают по переднему фронту синхросигнала (см. также рис. А.1).

А.5. На основании результатов выполнения пунктов А.3 и А.4, переходим к синтезу собственно УУ. Известны два класса управляющих автоматов: автоматы Мура и Мили [А.1]. Первые из них отличаются тем, что код *V*, управляющий ОБ, является функцией только от внутреннего состояния автомата (см. пункт А.6), в то время как у автоматов Мили он зависит также от кода операции, *Z* и от кода состояния ОБ, *U* (см. рис. А.1). Автоматы Мура характеризуются более высокой помехоустойчивостью, а автоматы Мили – меньшими затратами электронных компонентов или памяти на реализацию, что не является первостепенным преимуществом в настоящее время. Поэтому будем реализовывать синтезируемое УУ как автомат Мура.



ACC – аккумулятор
PC – программный счетчик
PM – память команд
IRG – регистр команды
F – выходы АЛУ
+ / L – вход управления режимом *PC* (суммирование / загрузка)
A – адресные входы *PM*
D – входы данных *ACC / PC / IRG*
Q – выходы *PC / PM / IRG*
C – входы тактирования *ACC / PC / IRG*
u0 – признак нулевого содержимого аккумулятора

Рис. А.2. Функциональная схема ОБ

А.6. Синтез начинаем с разработки **алгоритма** функционирования УУ. Введем понятие **состояния** УУ, под которым подразумевается определенный этап некоторой команды (см. пункт 2.6.4), выполняемый за один период тактового сигнала

ЦП. Переход из одного состояния в другое осуществляется по одному из фронтов (в рассматриваемом примере – по заднему) очередного синхроимпульса. Каждому состоянию автомата Мура соответствует определенное сочетание вырабатываемых УУ управляющих сигналов, называемое **микрокомандой**, по которой ОБ выполняет некоторую совокупность **микроопераций**, физически реализуемую параллельно во времени.

Начнем со словесного описания алгоритма работы синтезируемого УУ. На основании табл. А.1 и представленной на рис. А.2 функциональной схемы ОБ, выглядит следующим образом.

А.6.1. По включении питания УУ должно перейти в начальное состояние, которое обозначим *Q0*. В данном состоянии загрузка аккумулятора и регистра команды, а также изменение содержимого программного счетчика должны быть запрещены, т. е. значения битов *v1, v2* и *v3* должны быть равны нулю. Значения битов *v0* и *v4* при этом безразличны. Из состояния *Q0* должен быть выполнен безусловный переход в состояние «Выборка кода очередной команды», которое обозначим *Q1*. Предполагается, что по включении питания программный счетчик сбрасывается, т. е. по выходу из состояния *Q0* осуществляется выборка команды, расположенной по нулевому адресу памяти команд.

А.6.2. В состоянии *Q1* («Выборка кода очередной команды») загрузка аккумулятора и изменение содержимого программного счетчика должны быть запрещены, а загрузка регистра команды – разрешена, т. е. значения битов *v1* и *v3* должны быть равны нулю, а *v2* – единице. Значения битов *v0* и *v4* при этом безразличны. Состояние, в которое УУ должно перейти из состояния *Q1*, зависит от кода, загруженного в регистр команды (см. пункты А.7.3 – А.7.6).

А.6.3. Если код операции (т. е. комбинация 2-х старших битов кода команды) равен 00, из состояния *Q1* УУ должен перейти в состояние «Выполнение команды суммирования», которое обозначим *Q2*. В нем:

- АЛУ должно работать в режиме суммирования, т. е. значение бита *v0* должен быть равно нулю;

- запись в аккумулятор должна быть разрешена, т. е. бит *v1* должен быть в единичном состоянии;

- загрузка в регистр команды должна быть запрещена, т. е. значение бита *v2* должен быть нулевым;

- содержимое программного счетчика должно быть увеличено на единицу, т. к. после выполнения команды суммирования необходимо перейти к следующей за ней команде; поэтому в состоянии *Q2* значение бита *v3* должен быть равно единице, а *v4* – нулю.

Из состояния *Q2* должен быть выполнен безусловный переход к выборке очередной команды, т. е. в состояние *Q1*.

А.6.4. Если код операции равен 01, из состояния *Q1* УУ должен перейти в состояние «Выполнение команды вычитания», которое обозначим *Q3*. В нем:

- АЛУ должно работать в режиме вычитания, т. е. значение бита *v0* должен быть равно единице;

- значения битов *v1… v4* должны быть такими же, как и при выполнении команды суммирования (см. пункт А.7.3).

Из состояния *Q3* должен быть выполнен безусловный переход к выборке очередной команды, т. е. в состояние *Q1*.

А.6.5. Если код операции равен 10, что соответствует команде условного перехода по нулевому содержимому аккумулятора, а условие перехода не выполняется (*u0* = 0), из состояния *Q1* УУ должен перейти в состояние «Переход к следующей команде», которое обозначим *Q4*. В нем:

- запись в аккумулятор и в регистр команды должна быть запрещена, т. е. значения битов *v1* и *v2* должны быть равны нулю;

- состояние бита *v0* безразлично;

- содержимое программного счетчика должно быть увеличено на единицу, поэтому значение бита *v3* должен быть равно единице, а *v4* – нулю.

Из состояния *Q4* должен быть выполнен безусловный переход к выборке очередной команды, т. е. в состояние *Q1*.

А.6.6. Если код операции равен 10 (условный переход), и условие перехода выполняется (*u0* = 1), или указанная комбинация равна 11 (безусловный переход), из состояния *Q1* УУ должен перейти в состояние «Загрузка в программный счетчик адреса перехода», которое обозначим *Q5*. В нем:

- запись в аккумулятор и в регистр команды должна быть запрещена, т. е. биты *v1* и *v2* должны быть в нулевом состоянии;

- значение бита *v0* безразлично;

- в программный счетчик должен быть загружен адрес перехода, содержащийся в младших 6-и битах кода команды (см. рис. А.2), поэтому значения битов *v3* и *v4* должны быть равны единице.

Из состояния *Q5* должен быть выполнен безусловный переход к выборке очередной команды, т. е. в состояние *Q1*.

А.7. Наглядной и удобной формой изображения алгоритма работы управляющего автомата является его **диаграмма состояний** [А.1]. Она представляет собой ориентированный граф (граф переходов), вершины которого обозначают определенные состояния автомата, а ребра – переходы между состояниями.

Общие правила изображения диаграммы состояний автомата Мура следующие:

- на диаграмме должны быть отображены все возможные состояния УУ; каждое из них изображается как вершина графа переходов;

- рядом с каждой вершиной указываются коды управления ОБ, соответствующие состоянию УУ, обозначаемому данной вершиной;

- переходы между состояниями обозначаются стрелками, показывающими направления переходов; рядом со стрелками указываются условия переходов (если переход безусловный – они не указываются).

Диаграмма состояний синтезируемого УУ, составленная на основе приведенного в пункте А.6 словесного описания его работы, приведена на рис. А.3.

Примечание. Коды, управляющие ОБ, на рис. А.3 указаны в виде двоичных чисел формата *v4 v3 v2 v1 v0*.

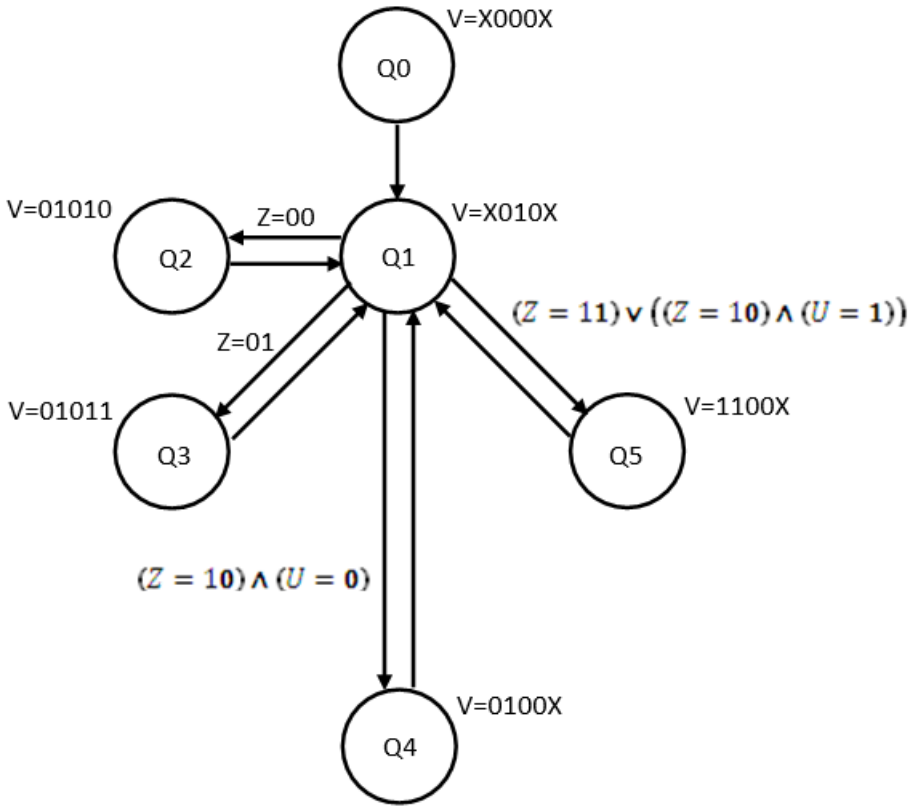


Рис. А.3. Диаграмма состояний синтезируемого УУ

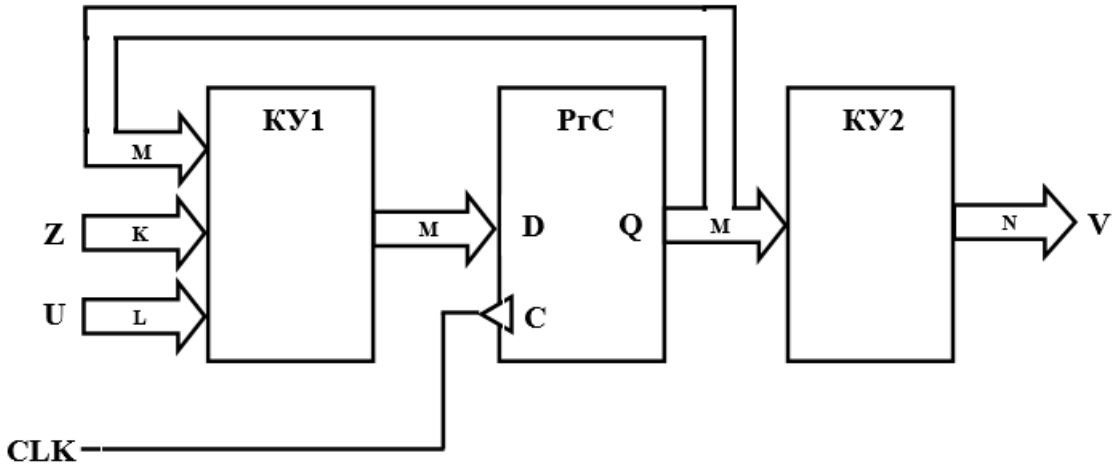
А.8. Техническая реализация алгоритма работы синтезируемого УУ может быть осуществлена 2-мя основными способами: в виде автомата с аппаратной («жесткой») логикой и с программируемой логикой [А.1]. Автоматы с аппаратной логикой предпочтительны при реализации УУ ЦП с *RISC*-архитектурой (относительно простой системой команд, ограниченной номенклатурой их форматов и способов адресации) [А.2], к которым относятся ЦП большинства современных семейств МК. Автоматы с программируемой логикой предпочтительнее для реализации УУ ЦП с *CISC*-архитектурой [А.2]. Поэтому будем рассматривать вариант реализации синтезируемого УУ как автомата с аппаратной логикой (тем более что данный вариант проще для пояснения). С принципами реализации автоматов с программируемой логикой можно ознакомиться, например, по источнику [А.2].

А.9. Аппаратная реализация автомата Мура осуществляется по обобщенной структурной схеме, приведенной на рис. А.4 [А.1]. Его основными блоками являются:

- параллельный, тактируемый фронтом (в рассматриваемом примере – задним) регистр кода текущего состояния УУ;

- комбинационное устройство КУ1, реализующее **функцию переходов** УУ, т. е. задающее, в какое состояние должно перейти УУ в следующем такте, в зависимости от его состояния в текущем такте, кода операции и состояния ОБ (см. рис. А.3);

- комбинационное устройство КУ2, реализующее **функцию выходов** УУ, т. е. формирующее код управления ОБ, как функцию от текущего состояния УУ.



РгС – регистр состояния УУ
КУ1, КУ2 – комбинационные устройства
K, L, M, N – разрядности кода операции, кодов состояния ОБ, состояния УУ и кода управления ОБ соответственно

Рис. А.4. Обобщенная структурная схема автомата Мура

Разрядности кода операции, кодов состояния ОБ и УУ, а также кода, управляющего ОБ, очевидно, зависят от реализуемой системы команд и от структуры ОБ. В рассматриваемом примере разрядность кода операции равна 2-м битам, кода состояния ОБ – одному биту, кода, управляющего ОБ – 5-и битам (см. табл. А.1 и рис. А.2). Разрядность регистра состояния УУ должна быть равна числу битов, необходимому для кодирования всех его возможных состояний, т. е. двоичному логарифму из их общего количества, округленному до ближайшего большего целого. В соответствии с рис. А.3, разрядность регистра состояния синтезируемого УУ должна быть

равна $\lceil log_2 6 \rceil$ (где $\lceil \cdot \rceil$ - оператор округления до ближайшего большего целого), т. е. 3-м битам.

Отметим, что состояние УУ изменяется только в моменты поступления активного фронта синхросигнала (в рассматриваемом примере - заднего). Следовательно, только в эти моменты изменяется и код управления, вырабатываемый автоматом Мура, в отличие от автомата Мили, генерируемый которым код управления может изменяться в произвольные моменты времени. Данное свойство автомата Мура обеспечивает его более высокую помехоустойчивость.

А.10. Дальнейший процесс синтеза УУ сводится к:

- выбору / разработке схемы регистра состояния УУ;
- синтезу КУ1 и КУ2.

Первая из перечисленных задач элементарна: регистр состояния реализуется по общеизвестной схеме параллельного регистра разрядностью *M* (в рассматриваемом примере равной 3-м), тактируемого синхросигналом УУ.

КУ1 и КУ2 синтезируются по общим правилам синтеза цифровых комбинационных устройств, на основе их таблиц истинности.

Входными сигналами КУ1 служат:

- состояния битов кода операции в текущем (i-м) такте, в рассматриваемом примере – младшего и старшего битов данного кода, которые обозначим *z0[i]* и *z1[i]*;
- код состояния ОБ в текущем такте, в рассматриваемом примере – уровень единственного оповещающего сигнала (см. рис. А.2), который обозначим *u0[i]*;
- уровни сигналов на выходах регистра состояния УУ в текущем такте, в рассматриваемом примере – 3-битового; обозначим данные уровни как *q0[i], q1[i]* и *q2[i]*.

На выходах КУ1 формируется **код перехода** УУ, представляющий собой код состояния, в которое должно перейти УУ в следующем такте при текущем сочетании кода операции, кодов состояния ОБ и УУ. Он подается на информационные входы регистра состояния, и по поступлении очередного синхроимпульса записывается в него. Тем самым обеспечиваются переходы в

соответствии с диаграммой состояний УУ. Обозначим состояния выходов КУ1 синтезируемого УУ в текущем такте (они же – состояния информационных входов РгС) как *d0[i], d1[i]* и *d2[i]*. При этом (см. выше) *d0[i] = q0[i+1], d1[i] = q1[i+1]* и *d2[i] = q2[i+1]*.

Входными переменными КУ2 служат уровни сигналов на выходах регистра состояния УУ. На выходах КУ2 формируется 5-битовый код, управляющий ОБ, т. е. **выходной код** УУ, у автомата Мура являющийся функцией от текущего состояния автомата (в рассматриваемом примере – УУ).

Кодирование состояний управляющего автомата (УУ) может осуществляться различными способами. Наиболее простой из них – представление состояний их номерами в двоичной системе счисления; остановимся на нем. Следует отметить, что в ряде конкретных случаев данный способ может быть не оптимален с точки зрения аппаратурных затрат на реализацию управляющего автомата, однако вопросы их оптимизации выходят за рамки излагаемого материала.

Таблицы истинности КУ1 и КУ2 составляются на основании диаграммы состояний автомата (УУ). Они обычно сводятся в общую таблицу, называемую **таблицей функционирования автомата**.

Таблица функционирования синтезируемого УУ, составленная на основе его диаграммы состояний (см. рис. А.3), представлена ниже. См. также **примечания** к ней.

Таблица А.2

Таблица функционирования синтезируемого УУ

Q[i]			**Z[i]**		**U[i]**	**D[i] = Q[i+1]**			**V[i]**				
q2[i]	**q1[i]**	**q0[i]**	**z1[i]**	**z0[i]**	**u0[i]**	**d2[i]**	**d1[i]**	**d0[i]**	**v4[i]**	**v3[i]**	**v2[i]**	**v1[i]**	**v0[i]**
1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	0	0	x	x	x	0	0	1	x	0	0	0	x
0	0	1	0	0	x	0	1	0	x	0	1	0	x
			0	1	x	0	1	1					
			1	0	0	1	0	0					
			1	0	1	1	0	1					
			1	1	x	1	0	1					
0	1	0	x	x	x	0	0	1	0	1	0	1	0
0	1	1							0	1	0	1	1
1	0	0							0	1	0	0	x
1	0	1							1	1	0	0	x

Окончание таблицы А.2

1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	1	0	x	x	x	0	0	0	x	0	0	0	x
1	1	1											

Примечания.

1. Кодами состояний УУ служат их номера в двоичной системе счисления.
2. Символом «x» обозначены безразличные состояния сигналов / переменных.
3. Состояния 110 (*Q6*) и 111 (*Q7*) являются «нештатными». При нормальной работе УУ переход в них исключен. Однако, он возможен, например, при сбоях, вызванных помехами. Поэтому при синтезе управляющих автоматов в таблицу их функционирования включаются и все возможные «нештатные» состояния. В них все сигналы, управляющие ОБ, должны быть не активны, и должен быть обеспечен безусловный переход из них в начальное состояние автомата (см. строки таблицы функционирования синтезируемого УУ, соответствующие состояниям 110 и 111).

А.11. На основании таблицы функционирования УУ выводятся логические выражения, описывающие зависимости состояний выходов КУ1 и КУ2 от состояний их входов. Вывод производится по известным методикам [А.2]; выражения могут быть представлены в виде совершенной дизъюнктивной или конъюнктивной нормальной формы (СДНФ или СКНФ соответственно). Для определенности, будем использовать представление в виде СДНФ.

В соответствии с табл. А.2, КУ1 синтезируемого УУ должен реализовывать следующую систему логических выражений:

$$d0[i] = \left(\overline{q2[i]} \wedge \overline{q1[i]} \wedge \overline{q0[i]}\right) \vee \left(\overline{q2[i]} \wedge \overline{q1[i]} \wedge q0[i] \wedge \overline{z1[i]} \wedge z0[i]\right) \vee$$
$$\vee \left(\overline{q2[i]} \wedge \overline{q1[i]} \wedge q0[i] \wedge z1[i] \wedge \overline{z0[i]} \wedge u0[i]\right) \vee$$
$$\vee \left(\overline{q2[i]} \wedge \overline{q1[i]} \wedge q0[i] \wedge z1[i] \wedge z0[i]\right) \vee \left(\overline{q2[i]} \wedge q1[i]\right) \vee$$
$$\vee \left(q2[i] \wedge \overline{q1[i]}\right);$$

$$d1[i] = \overline{q2[i]} \wedge \overline{q1[i]} \wedge q0[i] \wedge \overline{z1[i]};$$

$$d2[i] = \overline{q2[i]} \wedge \overline{q1[i]} \wedge q0[i] \wedge z1[i];$$

а КУ2 – следующую систему выражений:

$$v0[i] = \overline{q2[i]} \wedge q1[i] \wedge q0[i];$$

$$v1[i] = \overline{q2\,[i]} \wedge q1[i];$$

$$v2[i] = \overline{q2[i]} \wedge \overline{q1[i]} \wedge q0[i];$$

$$v3[i] = \left(\overline{q2[i]} \wedge q1[i]\right) \vee \left(q2[i] \wedge \overline{q1[i]}\right);$$

$$v4[i] = q2[i] \wedge \overline{q1[i]} \wedge q0[i].$$

На основе систем логических выражений, описывающих КУ1 и КУ2, осуществляется их схемная реализация. Принципы построения цифровых электронных устройств на базе их математических описаний подробно изложены во многих известных источниках, в том числе в [А.2], и останавливаться на них нет необходимости. Необходимо только отметить, что в настоящее время элементной базой УУ, естественно, служат логические элементы в интегральном исполнении, являющиеся компонентами структуры БИС МК.

А.12. Общие принципы синтеза и построения УУ реальных ЦП аналогичны описанным в пунктах А.1 – А.11. Отличия заключаются, в основном, в составе системы команд (на практике их не менее нескольких десятков) и функциональных узлов ЦП, количестве состояний цифрового автомата УУ и генерируемых им управляющих сигналов. В настоящее время синтез УУ и других блоков ЦП, естественно, не производится «вручную»; существует достаточно большой выбор средств автоматизации проектирования цифровых устройств.

Источники информации

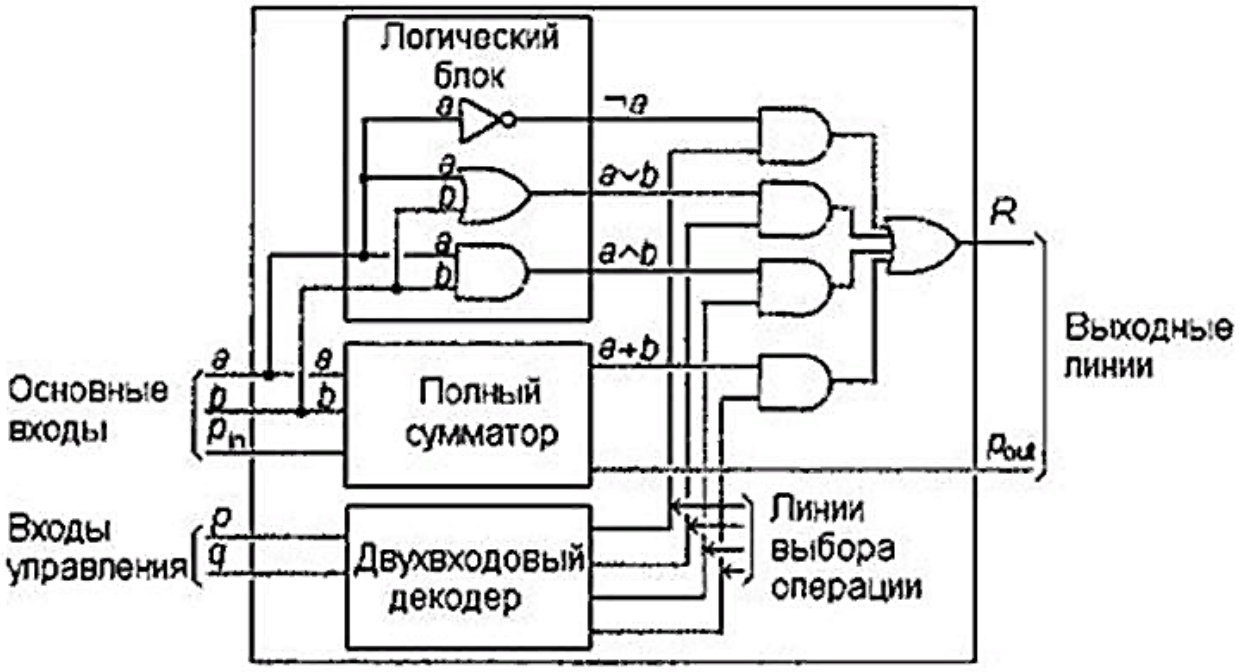
А.1. Каган Б.М. Электронные вычислительные машины и системы: Учеб. пособие для вузов. 3-е изд. – М.: Энергоатомиздат, 1991. – 592 с.

А.2. Орлов С. А., Цилькер Б. Я. Организация ЭВМ и систем: Учебник для вузов. 2-е изд. — СПб.: Питер, 2011. — 688 с.

Приложение Б. Простейший пример реализации АЛУ

АЛУ ЦП с относительно простой системой команд, в том числе АЛУ МК общего назначения, обычно реализуются в виде набора цифровых блоков, выполняющих элементарные арифметические и логические операции. Настройка АЛУ на выполнение той или иной операции осуществляется подключением выходов реализующего ее блока к выходам АЛУ, посредством мультиплексора, управляемого дешифратором команды.

Простейший пример построения АЛУ по данному принципу представлен на рис. Б.1 [Б.1]. Данное АЛУ реализует одну арифметическую операцию (суммирование) и три логических (инверсию, логическое ИЛИ и логическое И). Разрядность операндов – 1 бит.



p_{in} – вход переноса из предыдущего разряда
p_{out} – выход переноса в следующий разряд
R – выход АЛУ

Рис. Б.1. Пример функциональной схемы простейшего 1-битового АЛУ

АЛУ содержит:

- логический блок;

- арифметический блок, представляющий собой 1-битовый полный сумматор;

- декодер (дешифратор) кода операции с организацией «2 входа × 4 выхода»;

- мультиплексор на 4-х логических элементах 2И и одном элементе 2ИЛИ, управляемый дешифратором кода операции и осуществляющий подключение выхода реализующего ее блока к выходу АЛУ.

АЛУ с разрядностью операндов N бит ($N > 1$) может быть реализовано, например, объединением N 1-битовых АЛУ с соединенными между собой одноименными входами управления, а также с соединением выхода переноса каждого предыдущего разряда со входом переноса последующего.

По аналогичным общим принципам строятся реальные АЛУ; естественно, с намного более обширным набором операций и с использованием ряда приемов по минимизации времени их выполнения, а также аппаратурных затрат.

Источники информации

Б.1. Степанов А.Н. Архитектура вычислительных систем и компьютерных сетей. — СПб.: Питер, 2007. — 509 с.

Приложение В. Краткое руководство по расчету числовых значений кодов и битовых масок при программировании на регистровом уровне

Код, загружаемый в регистр, а также код маски при выборочной установке битов регистра в единицу (операцией поразрядного логического ИЛИ) рассчитывается по выражению:

$$x = \sum_{i=0}^{N-1} b_i \, w_i; \qquad (\text{В}.1)$$

где N – разрядность регистра, b_i – значение i – го бита кода (0 или 1), w_i – вес i – го бита при шестнадцатеричном представлении числа, в соответствии со 2-м столбцом табл. В.1.

Код маски при проверке состояний битов (операцией поразрядного логического И) также рассчитывается по выражению (В.1). При этом единичное значение присваивается только битам, состояние которых требуется проверить.

При расчете по выражению (В.1) в сумму включаются только веса битов, состояния которых должны быть установлены в единичное состояние или состояния которых необходимо проверить.

Код маски при выборочном обнулении битов регистра рассчитывается по следующему выражению:

$$x = \bigwedge_{i=0}^{N-1} \overline{b_i} m_i \, ; \qquad (\text{В}.2)$$

где $\wedge$ - оператор побитового логического И, $\overline{b_i}$ – инвертированное состояние i – го бита (равно 1 для битов, подлежащих обнулению, и 0 – для не подлежащих), m_i – код маски при обнулении i – го бита в соответствии с 3-м столбцом табл. В.1. При расчете по выражению (В.2) в логическое произведение включаются только коды маски битов, которые должны быть сброшены в нулевое состояние.

Таблица В.1

Веса битов при шестнадцатеричном представлении числа и коды битовых масок при обнулении битов

Номер бита	**Вес бита при шестнадцатеричном представлении числа (он же – битовая маска при установке бита в единицу или при проверке состояния бита)**	**Битовая маска при обнулении бита**
0	0x00000001	0xFFFFFFFE
1	0x00000002	0xFFFFFFFD
2	0x00000004	0xFFFFFFFB
3	0x00000008	0xFFFFFFF7
4	0x00000010	0xFFFFFFEF
5	0x00000020	0xFFFFFFDF
6	0x00000040	0xFFFFFFBF
7	0x00000080	0xFFFFFF7F
8	0x00000100	0xFFFFFEFF
9	0x00000200	0xFFFFFDFF
10	0x00000400	0xFFFFFBFF
11	0x00000800	0xFFFFF7FF
12	0x00001000	0xFFFFEFFF
13	0x00002000	0xFFFFDFFF
14	0x00004000	0xFFFFBFFF
15	0x00008000	0xFFFF7FFF
16	0x00010000	0xFFFEFFFF
17	0x00020000	0xFFFDFFFF
18	0x00040000	0xFFFBFFFF
19	0x00080000	0xFFF7FFFF
20	0x00100000	0xFFEFFFFF
21	0x00200000	0xFFDFFFFF
22	0x00400000	0xFFBFFFFF
23	0x00800000	0xFF7FFFFF
24	0x01000000	0xFEFFFFFF
25	0x02000000	0xFDFFFFFF
26	0x04000000	0xFBFFFFFF
27	0x08000000	0xF7FFFFFF
28	0x10000000	0xEFFFFFFF
29	0x20000000	0xDFFFFFFF
30	0x40000000	0xBFFFFFFF
31	0x80000000	0x7FFFFFFF

Примечание. Расчет как по выражению (В.1), так и (В.2) удобно производить, используя стандартное приложение «Калькулятор» ОС

Windows; опция «Программист», система представления чисел – *HEX* (шестнадцатеричная).

Примеры расчета кода по выражению (В.1).

1. Пусть необходимо установить в единичное состояние 13-й и 15-й биты младшего регистра конфигурации порта *A* МК *STM32F103xx*.

В соответствии с табл. В.1 вес 13-го бита равен 0x00002000, 15-го – 0x00008000. Их сумма равна *0x0000A000*, таков же, соответственно, и код маски.

Установка указанных битов в единичное состояние производится следующей командой уровня *CMSIS*:

GPIOA -> CRL |= 0x0000A000;

2. Пусть необходимо выполнить проверку состояния 2-го и 3-го битов регистра конфигурации подсистемы синхронизации (*RCC_CFGR*) МК *STM32F103xx*.

В соответствии с табл. В.1 вес 2-го бита равен 0x00000004, 3-го – 0x00000008. Их сумма равна *0x0000000C*, таков же, соответственно, и код маски. Проверка состояния битов осуществляется операцией (*RCC->CFGR*)&(*0x0000000C*). Значения 2-го и 3-го битов ее результата будут равны состояниям 2-го и 3-го битов регистра *RCC_CFGR*. Все остальные биты результата будут равны нулю.

Пример расчета кода по выражению (В.2).

Пусть необходимо обнулить биты с 4-го по 7-й младшего регистра конфигурации порта *A* МК *STM32F103xx*.

В соответствии с табл. В.1 коды маски вышеуказанных битов равны *0xFFFFFFEF, 0xFFFFFFDF, 0xFFFFFFBF* и *0xFFFFFF7F* соответственно. Результат их побитового логического И, а также, соответственно, код маски равен *0xFFFFFF0F*.

Обнуление указанных битов производится следующей командой уровня *CMSIS*:

GPIOA -> CRL &= 0xFFFFFF0F;