

Министерство образования Республики Беларусь

Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей
Кафедра ПОИТ

Дисциплина «Архитектура компьютерной техники и
операционных систем»

ОТЧЁТ
к лабораторным работам №2, №3

РАЗРАБОТКА ОПЕРАЦИОННОЙ ЧАСТИ И УПРАВЛЯЮЩЕГО
АВТОМАТА ДЛЯ ПРОСТЕЙШЕГО ЦИФРОВОГО
ВЫЧИСЛИТЕЛЬНОГО УСТРОЙСТВА

Вариант 8

Студент группы №351001
Ушаков А.Д.
Преподаватель
Леванцевич В.А.

Минск 2024

СОДЕРЖАНИЕ

Введение.....	3
1 Граф-схема алгоритма	4
1.1 Схема алгоритма.....	4
1.2 Описание алгоритма.....	5
Вывод по главе 1 «Граф-схема алгоритма».....	5
2 Операционная часть	6
2.1 Структурная схема операционной части	6
2.2 Принципиальная схема операционной части	7
Вывод по главе 2 «Операционная часть»	7
3 Управляющий автомат.....	8
3.1 Разработка микропрограммы.....	8
3.2 Нех-файлы.....	8
3.3 Принципиальная схема управляющего автомата	9
Вывод по главе 3 «Управляющий автомат».....	9
Заключение	11
Список использованных источников	13

ВВЕДЕНИЕ

Большинство цифровых вычислительных устройств состоит, как правило, из трех частей: запоминающей, операционной части и управляющей части (см. рисунок 1).

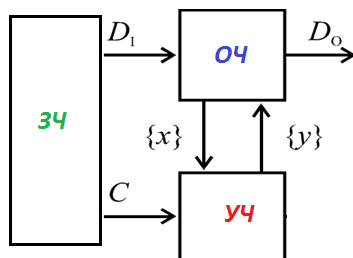


Рисунок 1 – Структура вычислительного устройства

Операционная часть выполняет обработку цифровых данных под управлением управляющей части (управляющего автомата).

При разработке управляющих автоматов используются понятия абстрактного и структурного автомата.

Абстрактный автомат (см. рисунок 2) это математическая модель не учитывающая его внутреннюю структуру и описывает только поведение автомата во времени. Сама структура автомат не известна.

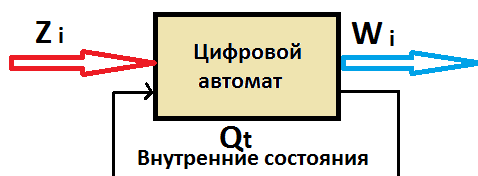


Рисунок 2 – Абстрактный автомат

Структурный автомат представляет конкретную реализацию управляющего автомата, в зависимости от его архитектуры и заданного набора элементов используемых для его реализации.

В варианте №8 необходимо реализовать умножение с младших разрядов множителя со сдвигом суммы частичных произведений вправо в качестве операционной части и синтезировать управляющий автомат на основе микропрограмм (МП).

1 ГРАФ-СХЕМА АЛГОРИТМА

1.1 Схема алгоритма

На рисунке 1.1 представлена схема алгоритма умножения с младших разрядов множителя со сдвигом суммы частичных произведений вправо.

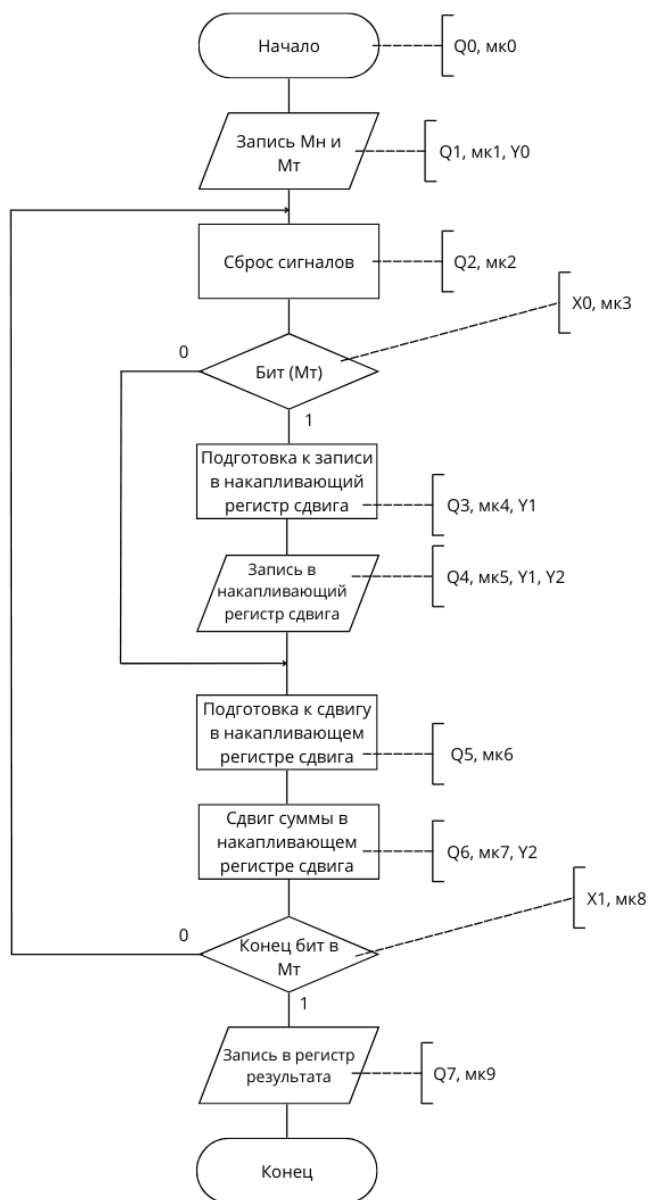


Рисунок 1.1 – Схема алгоритма умножения с младших разрядов множителя со сдвигом суммы частичных произведений вправо

2 ОПЕРАЦИОННАЯ ЧАСТЬ

2.1 Структурная схема операционной части

На рисунке 2.1 представлена структурная схема умножения с младших разрядов множителя со сдвигом суммы частичных произведений вправо.

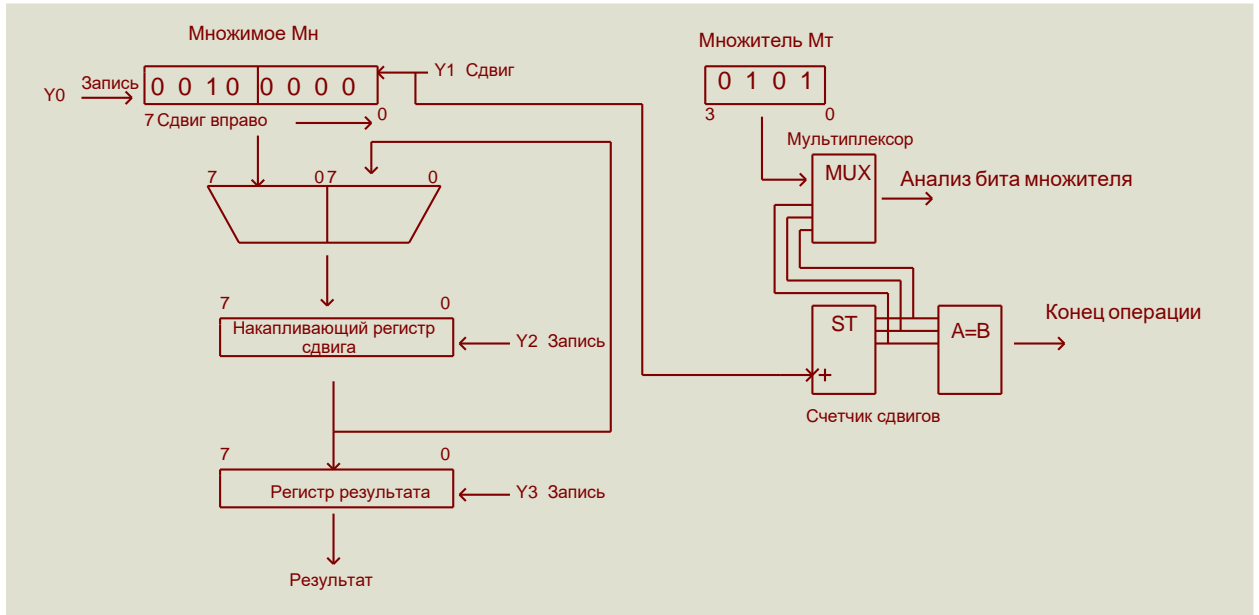


Рисунок 2.1 – Структурная схема умножения с младших разрядов множителя со сдвигом суммы частичных произведений вправо

2.2 Принципиальная схема операционной части

На рисунке 2.2 представлена принципиальная схема умножения с младших разрядов множителя со сдвигом суммы частичных произведений вправо.

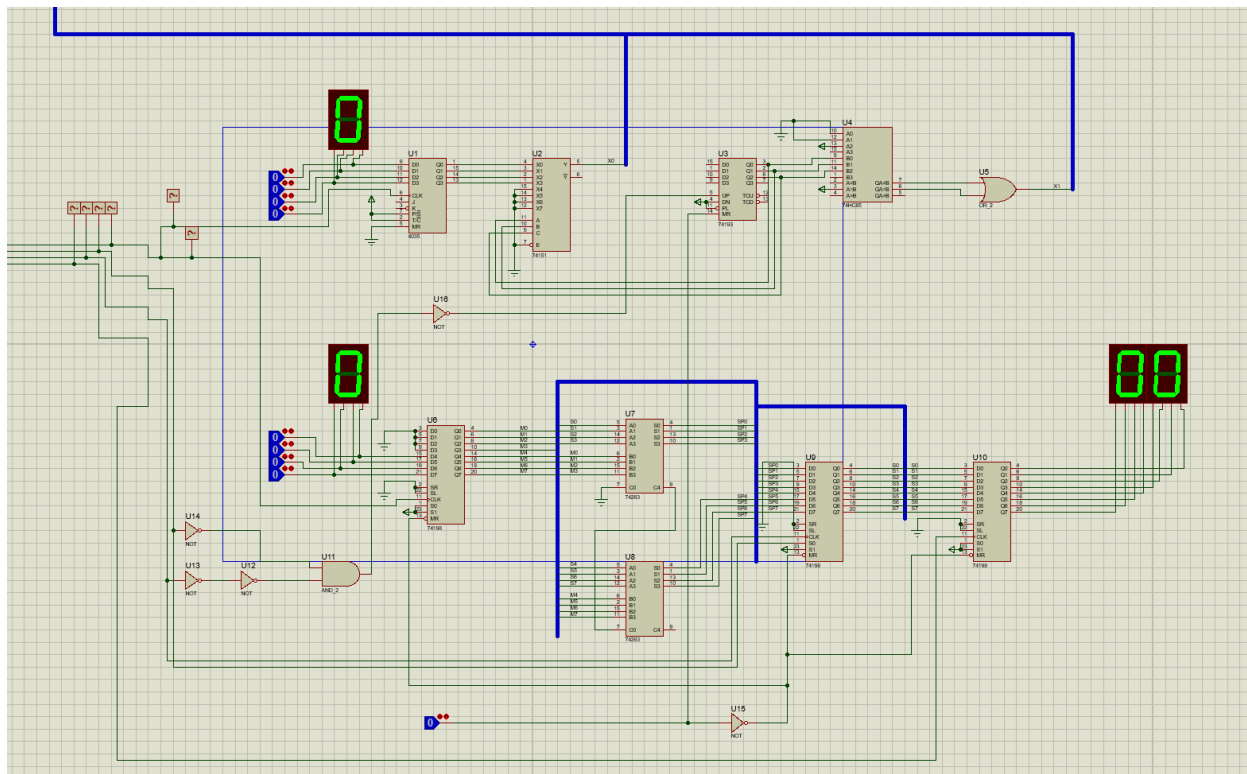


Рисунок 2.2 – Принципиальная схема умножения с младших разрядов множителя со сдвигом суммы частичных произведений вправо

Вывод по главе 2 «Операционная часть»

Операция умножения состоит из ряда последовательных сложений. Сложением управляют разряды множителя: если в очередном разряде множителя содержится единица, то к сумме добавляется множимое. При этом, в зависимости от метода умножения, выполняется сдвиг либо множимого, либо частичной суммы. Наряду с этим умножение можно начинать как с младших, так и со старших разрядов множителя. Рассмотрим правила умножения в прямых кодах без учета знака.

3 УПРАВЛЯЮЩИЙ АВТОМАТ

3.1 Разработка микропрограммы

В таблице 3.1 представлена микропрограмма, в которой отражена связь каждой микрокоманды с текущим и следующим состояниями МПЗУ.

Таблица 3.1 – Микропрограмма

№ МК	Тек. сост.	Адрес на входе ПЗУ				Адрес следующей МК				N _x		V	hex	Выходные сигналы			
										N _{x1}	N _{x2}			Y ₃	Y ₂	Y ₁	Y ₀
МК ₀	Q ₀	0	0	0	0	0	0	0	1	1	1	1	0F	0	0	0	0
МК ₁	Q ₁	0	0	0	1	0	0	1	0	1	1	1	17	0	0	0	1
МК ₂	Q ₂	0	0	1	0	0	0	1	1	1	1	1	1F	0	0	0	0
МК ₃	Q ₂	0	0	1	1	0	1	1	0	0	1	0	32	0	0	0	0
МК ₄	Q ₃	0	1	0	0	0	1	0	1	1	1	1	2F	0	0	1	0
МК ₅	Q ₄	0	1	0	1	0	1	1	0	1	1	1	37	0	1	1	0
МК ₆	Q ₅	0	1	1	0	0	1	1	1	1	1	1	3F	0	0	0	0
МК ₇	Q ₆	0	1	1	1	1	0	0	0	1	1	1	47	0	1	0	0
МК ₈	Q ₆	1	0	0	0	0	0	1	0	1	0	0	14	0	1	0	0
МК ₉	Q ₇	1	0	0	1	1	0	1	0	1	1	1	57	1	0	0	0
БП	Q ₇	1	0	1	0	0	0	0	1	0	0	0	08	1	0	0	0

3.2 Нех-файлы

На рисунке 3.1 представлены hex-файлы для первого ПЗУ, которое фиксирует адреса микрокоманд, и для второго ПЗУ, которое фиксирует выходные сигналы.

```

:010000000FF0 :0100000000FF
:0100010017E7 :0100010001FD
:010002001FDE :0100020000FD
:0100030032CA :0100030000FC
:010004002FCC :0100040002F9
:0100050037C3 :0100050006F4
:010006003FBA :0100060000F9
:0100070047B1 :0100070004F4
:0100080014E3 :0100080004F3
:01000900579F :0100090008EE
:01000A0008ED :01000A0008ED
:00000001FF| :00000001FF

```

Рисунок 3.1 – ПЗУ1 и ПЗУ2

3.3 Принципиальная схема управляющего автомата

На рисунке 3.2 представлена принципиальная схема управляющего автомата для операционной части умножения с младших разрядов множителя со сдвигом суммы частичных произведений вправо.

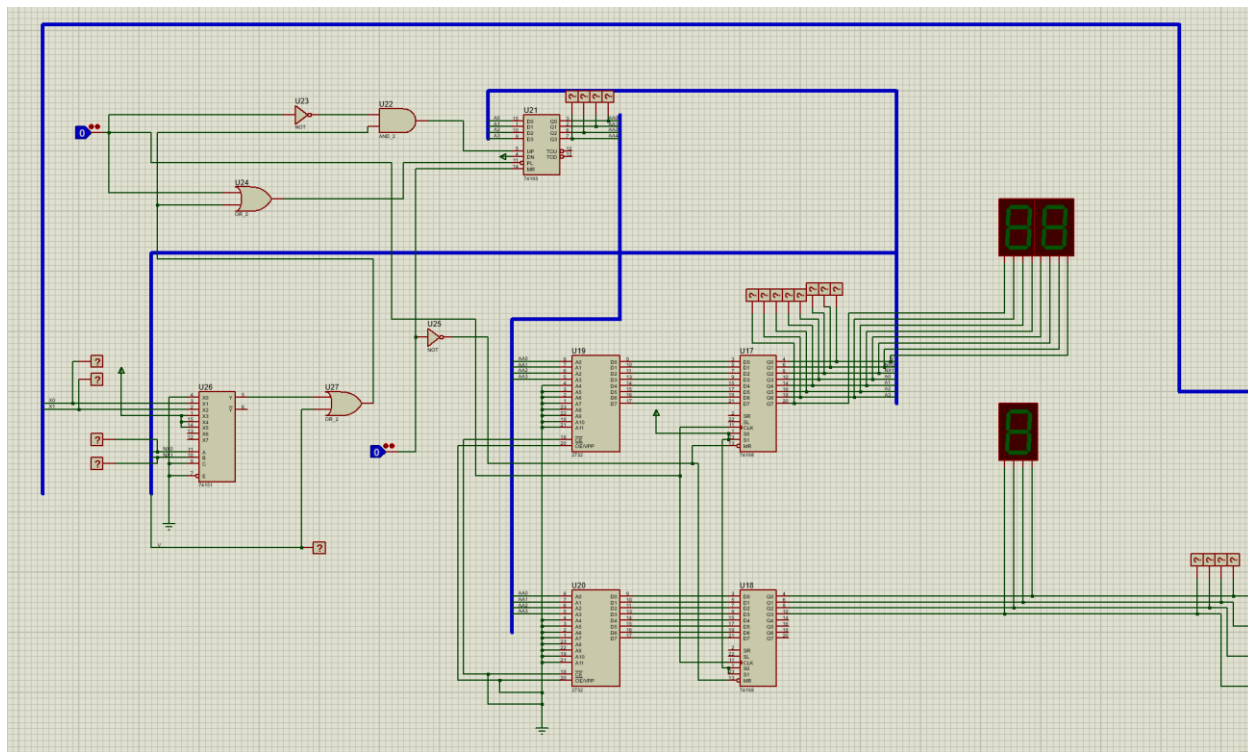


Рисунок 3.2 – Принципиальная схема управляющего автомата

Вывод по главе 3 «Управляющий автомат»

Микропрограмма записана в микропрограммное постоянное запоминающее устройство МПЗУ. Каждая микрокоманда состоит из четырех полей **V**: **A** : **Nx** : **Y**.

Существует два типа микрокоманд, различающихся старшим битом **V**:

V=1 – ОМК: операционная микрокоманда, содержащая в поле **Y** значения бит соответствующие сигналам **Y1 - Yn**.

V=0 – УМК: управляющая микрокоманда. По этой команде происходит переход на микрокоманду адрес которой хранится в поле **A** управляющей команды.

Поле **A** – адрес перехода, подаётся на входы предзагрузки данных **D** регистра счётчика адреса микрокоманд **РАМК** и в зависимости от значения анализируемой переменной **Xi**, поступающей с мультиплексора **MS**, формирует адрес перехода.

N_x – закодированный номер проверяемой входной переменной. Это значение подается на адресные входы мультиплексора **MS** и разрешает прохождение на его выход значения соответствующей переменной X_i . Безусловный переход реализуется путем фиксирования лог. 0 на первом входе данных MS. Т.е. автомат всегда выполняет переход, если $X_i=0$.

$Y_1 - Y_n$ – управляющие сигналы формируемые управляющим автоматом.

Счётчик-РАМК при $V=0$ выполняет загрузку с входов D адреса следующей команды (для команды перехода). При $V=1$ выполняется счёт (инкремент – увеличение значения на 1) по тактовым импульсам на входе +1, по которым происходит выполнение микрокоманд по последовательным адресам.

РМК (регистр микрокоманды) осуществляет промежуточное хранение микрокоманды на время её выполнения.

Функционирование автомата состоит в следующем.

Если автомат выполняет ОМК, то 1 ($V=1$) из старшего бита МК блокирует мультиплексор входных сигналов MS элементом ИЛИ и запрещает счётчику-РАМК выполнять предзагрузку адреса перехода. На выходе ПЗУ формируются управляющие сигналы Y_i . По приходу очередного тактового импульса счётчик увеличится на 1, последовательно выбирая следующую МК.

Если выполняется УМК ($V=0$), то выходные значения Y_i сохраняют предыдущее значение, а элемент ИЛИ разрешает прохождение сигнала данных с мультиплексора. Далее, если $X_i=0$, то прохождение тактовых импульсов на счётчик запрещается, но разрешается предзагрузка значением поля адреса А микрокоманды и выполняется переход. Если $X_i=1$, то предзагрузка не происходит, а разрешен проход тактового импульса на счетный вход +1, в результате перехода по адресу нет, и счетчик последовательно формирует по следующий адрес микрокоманды.

ЗАКЛЮЧЕНИЕ

В процессе изучения основ цифровой электроники и обработки сигналов важно понимать правила выполнения операций в двоичном коде, которые являются основой работы с цифровыми системами. Двоичная система исходит из двух состояний, 0 и 1, и операции, такие как сложение, вычитание, умножение и деление, выполняются по специфическим правилам, аналогичным арифметическим операциям в десятичной системе, но с использованием двоичных разрядов. Эти правила позволяют эффективно обрабатывать и хранить данные в цифровом формате.

Временные диаграммы и последовательность управляющих сигналов играют ключевую роль в синхронизации работы цифровых систем. Временные диаграммы визуализируют изменения сигналов во времени, что помогает разработчикам и инженерам понять, как различные компоненты взаимодействуют друг с другом. Последовательность управляющих сигналов определяет порядок операций и взаимодействие между устройствами, обеспечивая правильное выполнение задач и минимизируя вероятность ошибок в системе.

Создание схемы-алгоритма собственного метода позволяет визуализировать и формализовать процесс решения конкретной задачи. Такая схема помогает структурировать мысли и представить последовательность действий, необходимых для достижения результата, что особенно полезно при разработке сложных систем или алгоритмов.

Дополнительно следует рассмотреть назначение и принцип работы аналогово-цифровых преобразователей (АЦП) и цифрово-аналоговых преобразователей (ЦАП). АЦП преобразует аналоговые сигналы в цифровую форму, что позволяет обрабатывать их с использованием цифровых устройств. ЦАП, наоборот, выполняет обратную задачу, преобразуя цифровые данные обратно в аналоговые сигналы, что необходимо для взаимодействия с аналоговыми устройствами. Эти преобразователи играют важную роль в таких областях, как обработка звука, изображений и измерение физических величин.

Теорема Котельникова-Найквиста, также известная как теорема о дискретизации, устанавливает условия для точного восстановления сигнала из его дискретных значений. Она утверждает, что для того, чтобы восстановить сигнал без потерь, он должен быть дискретизирован с частотой, как минимум вдвое превышающей максимальную частоту данного сигнала. Это принципиально важно для цифровой обработки сигналов, обеспечивая высокое качество и точность воспроизведения информации.

В ходе изучения управляющих автоматов были рассмотрены ключевые понятия и структуры, которые лежат в основе их функционирования. Абстрактный управляющий автомат представляет собой математическую

модель, описывающую системы, которые принимают входные сигналы и формируют выходные действия в зависимости от текущего состояния и входных данных. Структурный управляющий автомат детализирует архитектуру и внутренние компоненты, позволяя лучше понять взаимодействие между различными элементами автомата.

Структурная схема управляющего автомата Мили демонстрирует, что выходные сигналы зависят как от текущего состояния, так и от входных сигналов, что придаёт модели гибкость в управлении. В отличие от этого, структурная схема управляющего автомата Мура показывает, что выходные сигналы зависят только от текущего состояния, что делает автомат более предсказуемым. Структурная схема микропрограммного автомата представляет собой более сложную архитектуру, в которой используются микрокоманды для управления процессами, включая управляющие регистры и логические элементы, что обеспечивает высокую степень контроля и гибкости.

Сравнительный анализ управляющих автоматов показывает, что автоматы Мили более универсальны, так как их выходы могут изменяться в зависимости от входов, тогда как автоматы Мура предлагают более простую и стабильную логику работы. Микропрограммные автоматы предоставляют высокий уровень детализации и адаптации к сложным задачам. Формат hex-файла, используемый для представления двоичных данных в текстовом виде, состоит из строк, каждая из которых начинается с символа : и содержит адрес, тип данных и контрольную сумму, что позволяет легко обмениваться и загружать данные в микроконтроллеры и другие устройства.

Эти аспекты управления автоматами играют важную роль в разработке современных систем, обеспечивая эффективное управление и обработку данных в различных приложениях.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

[1] Леванцевич, В. А. Разработка операционной части простейшего цифрового вычислительного устройства / В. А. Леванцевич // Лабораторные работы. – 2024. – №2.

[2] Леванцевич, В. А. Разработка цифрового управляющего автомата / В. А. Леванцевич // Лабораторные работы. – 2024. – №3.