



[www.maxim-ic.com](http://www.maxim-ic.com)

## ОПИСАНИЕ

DS18B20 цифровой термометр с программируемым разрешением, от 9 до 12-bit, которое может сохраняться в EEPROM памяти прибора. DS18B20 обменивается данными по 1-Wire шине и при этом может быть как единственным устройством на линии так и работать в группе. Все процессы на шине управляются центральным микропроцессором.

Диапазон измерений от  $-55^{\circ}\text{C}$  до  $+125^{\circ}\text{C}$  и точностью  $0.5^{\circ}\text{C}$  в диапазоне от  $-10^{\circ}\text{C}$  до  $+85^{\circ}\text{C}$ . В дополнение, DS18B20 может питаться напряжением линии данных ("parasite power"), при отсутствии внешнего источника напряжения.

Каждый DS18B20 имеет уникальный 64-битный последовательный код, который позволяет, общаться с множеством датчиков DS18B20 установленных на одной шине. Такой принцип позволяет использовать один микропроцессор, чтобы контролировать множество датчиков DS18B20, распределенных по большому участку. Приложения, которые могут извлечь выгоду из этой особенности, включают системы контроля температуры в зданиях, и оборудовании или машинах, а так же контроль и управление температурными процессами.

Назначение выводов Table 1

SO*	SOP*	TO-92	СИМВОЛ	ОПИСАНИЕ
5	4	1	GND	Общий.
4	1	2	DQ	Вывод данных ввода/вывода (Input/Output pin). Open-drain 1-Wire interface pin. По этой линии подается питание в режиме работы с паразитным питанием.
3	8	3	VDD	VDD ножка питания. Для режима работы с паразитным питанием VDD необходимо соединить с общим проводом.

\*Все остальные выводы должны оставаться не подключенными.

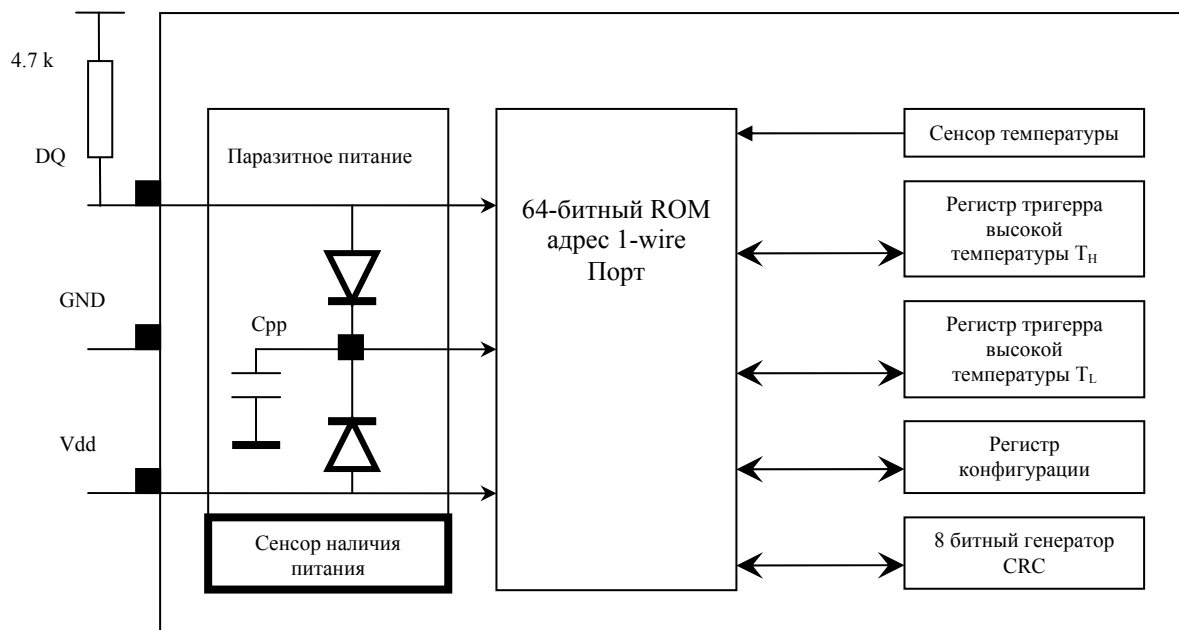
## КРАТКИЙ ОБЗОР

Иллюстрация 1 показывает блок-схему DS18B20, и описания выводов даются в Таблице 1. 64-битовый ROM запоминает уникальный последовательный код прибора. Оперативная память содержит 2-байтовый температурный регистр, который запоминает цифровой выход от температурного преобразователя. Кроме того, электронный блок обеспечивает доступ к 1-байтовым сигнальным регистрам триггерной схемы (TH и TL), и к регистру конфигурации. Регистр конфигурации позволяет пользователю устанавливать разрешающую способность цифрового преобразователя температуры к 9, 10, 11, или 12 битам. TH, TL и регистры конфигурации энергонезависимы (EEPROM), таким образом они сохраняют данные, когда прибор - выключен.

DS18B20 использует исключительно 1-Wire протокол – при этом формируется соединение, которое осуществляет коммуникацию на шине, используя всего один управляющий сигнал. Шина должна быть подключена к источнику питания через подтягивающий резистор, так как все приборы связаны с шиной, используют соединение через Z-состояния или вход открытого стока. Используя эту шину микропроцессор (ведущий) идентифицирует и обращается к датчикам температуры, используя 64-битовый код прибора. Поскольку каждый прибор имеет уникальный код, число приборов, к которым можно обратиться на одной шине, фактически неограниченно.

Другая особенность DS18B20 - способность работать без внешнего питания. Эта возможность предоставляется через подтягивающий резистор. Высокий сигнал шины заряжает внутренний конденсатор (C<sub>PP</sub>), который питает прибор, когда на шине низкий уровень. Этот метод носит название «Паразитное питание». При этом максимальная измеряемая температура при этом  $+100^{\circ}\text{C}$ . Для расширения диапазона температур до  $+125^{\circ}\text{C}$  необходимо использовать внешнее питание.

Рисунок -2



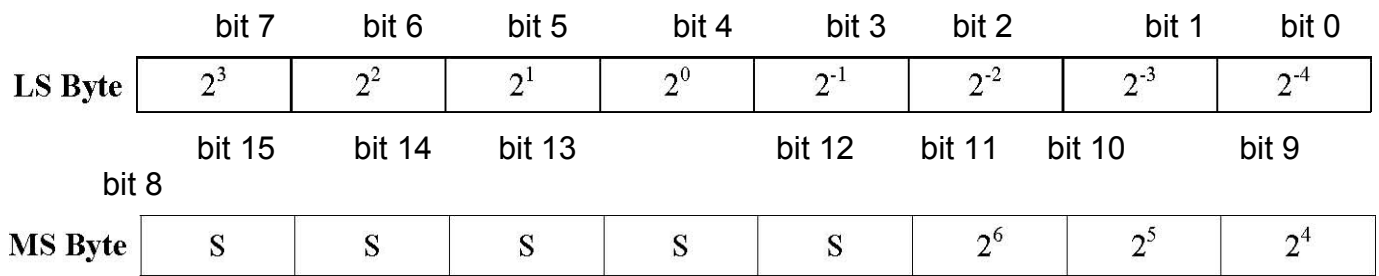
### РЕЖИМ – конвертации температуры

Основные функциональные возможности DS18B20 - его температурный преобразователь. Разрешающая способность температурного преобразователя может быть изменена пользователем и составляет 9, 10, 11, или 12 битов, соответствуя приращениям (дискретности измерения температуры) 0.5 °C, 0.25°C, 0.125°C, и 0.0625°C, соответственно. Разрешающая способность по умолчанию установлена 12-бит. В исходном состоянии DS18B20 находится в состоянии покоя ( в неактивном состоянии). Чтобы начать температурное измерение и преобразование, ведущий должен подать команду начала конвертирования температуры [0x44]. После конвертирования, полученные данные запоминаются в 2-байтовом регистре температуры в оперативной памяти, и DS18B20 возвращается к его неактивному состоянию. Если DS18B20 включен с внешним питанием, ведущий может контролировать конвертирование температуры (после команды [0x44]) по состоянию шины. **На шине будет присутствовать логический «0» когда происходит температурное преобразование. И логическая «1», когда конвертирование выполнено.** Если DS18B20 включен с паразитным питанием, эта технология уведомления не может быть использована, так как шину нужно подать высокий уровень (напряжение питания) в течение всего времени температурного преобразования.

Выходные температурные данные DS18B20 калиброваны в градусах Цельсия. Температурные данные запоминаются как 16-битовое число со знаком (см. иллюстрацию 2). Биты признака (S) указывают, является ли температура положительная или отрицательная: для положительных  $S = 0$ , а для отрицательных чисел  $S = 1$ . Если DS18B20 будет настроен для конвертирования 12-битной разрешающей способности, то все биты в температурном регистре будут содержать действительные данные. Для 11-битной разрешающей способности, бит 0 неопределен. Для 10-битной разрешающей способности, биты 1 и 0 неопределены, и для 9 битной разрешающей способности 2, 1 и 0 неопределены.

Таблица 2 дает примеры данных цифрового выхода и соответствующей температуры, для 12- битной разрешающей способности.

## TEMPERATURE REGISTER FORMAT Figure 2



TEMPERATURE/DATA RELATIONSHIP Table 2

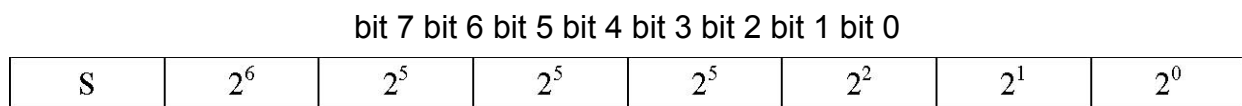
Температура	Цифровой выход (двоичный)	Цифровой выход (Hex)
+125°C	0000 0111 1101 0000	07D0h
+85°C*	0000 0101 0101 0000	0550h
+25.0625°C	0000 0001 1001 0001	0191h
+10.125°C	0000 0000 1010 0010	00A2h
+0.5°C	0000 0000 0000 1000	0008h
0°C	0000 0000 0000 0000	0000h
-0.5°C	1111 1111 1111 1000	FFF8h
-10.125°C	1111 1111 0101 1110	FF5Eh
-25.0625°C	1111 1110 0110 1111	FE6Fh
-55°C	1111 1100 1001 0000	FC90h

\*при подаче питания в температурный регистр записано число +85°C

## OPERATION — ALARM SIGNALING

After the DS18B20 performs a temperature conversion, the temperature value is compared to the user-defined two's complement alarm trigger values stored in the 1-byte TH and TL registers (see Figure 3). The sign bit (S) indicates if the value is positive or negative: for positive numbers  $S = 0$  and for negative numbers  $S = 1$ . The TH and TL registers are nonvolatile (EEPROM) so they will retain data when the device is powered down. TH and TL can be accessed through bytes 2 and 3 of the scratchpad as explained in the MEMORY section of this datasheet.

## TH AND TL REGISTER FORMAT Figure 3



Only bits 11 through 4 of the temperature register are used in the TH and TL comparison since TH and TL are 8-bit registers. If the measured temperature is lower than or equal to TL or higher than TH, an alarm condition exists and an alarm flag is set inside the DS18B20. This flag is updated after every temperature measurement; therefore, if the alarm condition goes away, the flag will be turned off after the next temperature conversion.

The master device can check the alarm flag status of all DS18B20s on the bus by issuing an Alarm Search [ECh] command. Any DS18B20s with a set alarm flag will respond to the command, so the master can determine exactly which DS18B20s have experienced an alarm condition. If an alarm condition exists and the TH or TL settings have changed, another temperature conversion should be done to validate the alarm condition.

## POWERING THE DS18B20

DS18B20 может быть включен с внешним питанием VDD, или он может работать в режиме «паразитного питания», которое позволяет DS18B20 функционировать без питания на выводе VDD. Паразитное питание очень полезно для приложений, которые требуют отдаленного температурного считывания, или это ограничение связано со старыми линиями коммуникаций, где уже проложено только два провода. Иллюстрация 1 показывает схеме управления власти паразита DS18B20's, которая "захватывает" власть от шины C 1 проводом через способный к глубокой вытяжке штырек, когда шина высока. Захваченные степени обвинения, которые DS18B20, в то время как шина высока, и часть обвинения, сохранен на конденсаторе власти паразита (CPP), чтобы обеспечить власти, когда шина низка. Чтобы DS18B20 использовать в режиме паразитного питания, вывод VDD должен быть связан VSS.

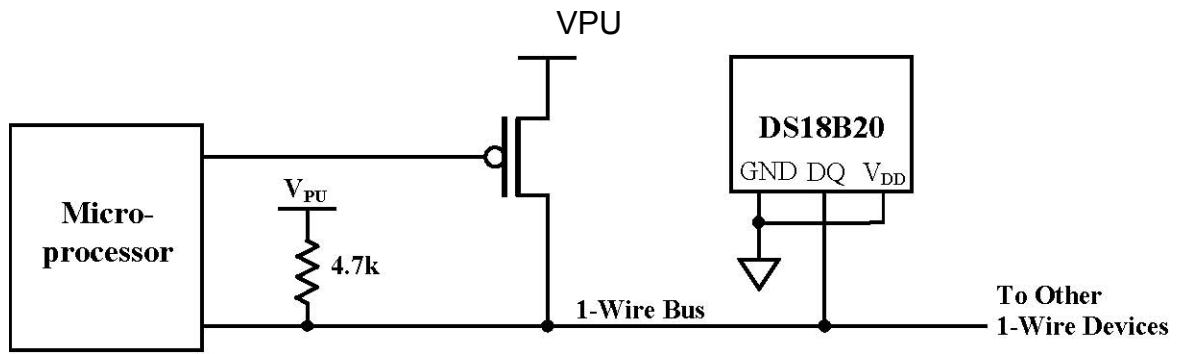
In parasite power mode, the 1-Wire bus and CPP can provide sufficient current to the DS18B20 for most operations as long as the specified timing and voltage requirements are met (refer to the DC ELECTRICAL CHARACTERISTICS and the AC ELECTRICAL CHARACTERISTICS sections of this data sheet). However, when the DS18B20 is performing temperature conversions or copying data from the scratchpad memory to EEPROM, the operating current can be as high as 1.5mA. This current can cause an unacceptable voltage drop across the weak 1-Wire pullup resistor and is more current than can be supplied by CPP. To assure that the DS18B20 has sufficient supply current, it is necessary to provide a strong pullup on the 1-Wire bus whenever temperature conversions are taking place or data is being copied from the scratchpad to EEPROM. This can be accomplished by using a MOSFET to pull the bus directly to the rail as shown in Figure 4. The 1-Wire bus must be switched to the strong pullup within 10µs (max) after a Convert T [44h] or Copy Scratchpad [48h] command is issued, and the bus must be held high by the pullup for the duration of the conversion (tconv) or data transfer (twr = 10ms). No other activity can take place on the 1-Wire bus while the pullup is enabled.

The DS18B20 can also be powered by the conventional method of connecting an external power supply to the VDD pin, as shown in Figure 5. The advantage of this method is that the MOSFET pullup is not required, and the 1-Wire bus is free to carry other traffic during the temperature conversion time.

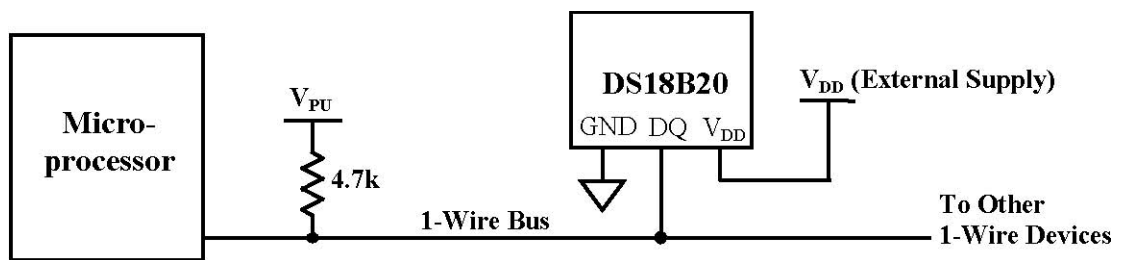
При использовании паразитного питания не рекомендуется использовать датчики для температуры выше +100°C since the DS18B20 may not be able to sustain communications due to the higher leakage currents that can exist at these temperatures. For applications in which such temperatures are likely, it is strongly recommended that the DS18B20 be powered by an external power supply.

In some situations the bus master may not know whether the DS18B20s on the bus are parasite powered or powered by external supplies. The master needs this information to determine if the strong bus pullup should be used during temperature conversions. To get this information, the master can issue a Skip ROM [CCh] command followed by a Read Power Supply [B4h] command followed by a "read time slot". During the read time slot, parasite powered DS18B20s will pull the bus low, and externally powered DS18B20s will let the bus remain high. If the bus is pulled low, the master knows that it must supply the strong pullup on the 1-Wire bus during temperature conversions.

SUPPLYING THE PARASITE-POWERED DS18B20 DURING TEMPERATURE CONVERSIONS Figure 4



POWERING THE DS18B20 WITH AN EXTERNAL SUPPLY Figure 5



## 64-BIT LASERED ROM CODE

Каждый DS18B20 содержит уникальный 64-битовый код (см. иллюстрацию 6), сохраненный в ROM. Младшие 8 битов кода ROM содержат код семейства 1-Wire проводом DS18B20's: 28h. Следующие 48 битов, содержат уникальный серийный номер. Старшие 8 битов содержат циклический контроль избыточности (CRC) байт, который вычислен от первых 56 битов кода ROM. Детализированное объяснение битов циклического контроля избыточности обеспечивается в разделе GENERATION циклического контроля избыточности. 64-битовый код ROM и связанная функция ROM управляют логикой, позволяют DS18B20 работать как 1-Wire устройство, используя протокол, детализированный в разделе 1-WIRE BUS SYSTEM этой таблицы данных.

## 64-BIT LASERED ROM CODE Figure 6

8-BIT CRC	48-BIT SERIAL NUMBER	8-BIT FAMILY CODE (28h)
	MSB LSB MSB LSB MSB LSB	

## MEMORY

Память DS18B20's организована, как показано в иллюстрации 7. Память состоит из сверхоперативной SRAM с энергонезависимой памятью EEPROM. Первые два регистра это регистры конвертора температуры, далее идет сигнальный триггер тревоги с регистрами верхнего и нижнего предела (T<sub>H</sub> and T<sub>L</sub>) и регистр конфигурации. Обратите внимания, что, если функция тревоги DS18B20 не используется, T<sub>H</sub> and T<sub>L</sub> могут служить универсальной памятью. Все команды памяти описаны подробно в разделе FUNCTION COMMANDS DS18B20.

Байт 0 и байт 1 из сверхоперативных содержит младший бит и MSB температурного регистра, соответственно. Эти байты только для чтения. Байты 2 и 3 обеспечивают доступ к регистрам T<sub>H</sub> and T<sub>L</sub>. Байт 4 содержит данные регистра конфигурации, которые объясняются подробно в разделе РЕГИСТРА КОНФИГУРАЦИИ (CONFIGURATION REGISTER) этой таблицы данных. Байты 5, 6, и 7 зарезервированы для внутреннего использования устройством и запись в эти регистры невозможна; при чтении эти байты возвратят «1» во всех разрядах.

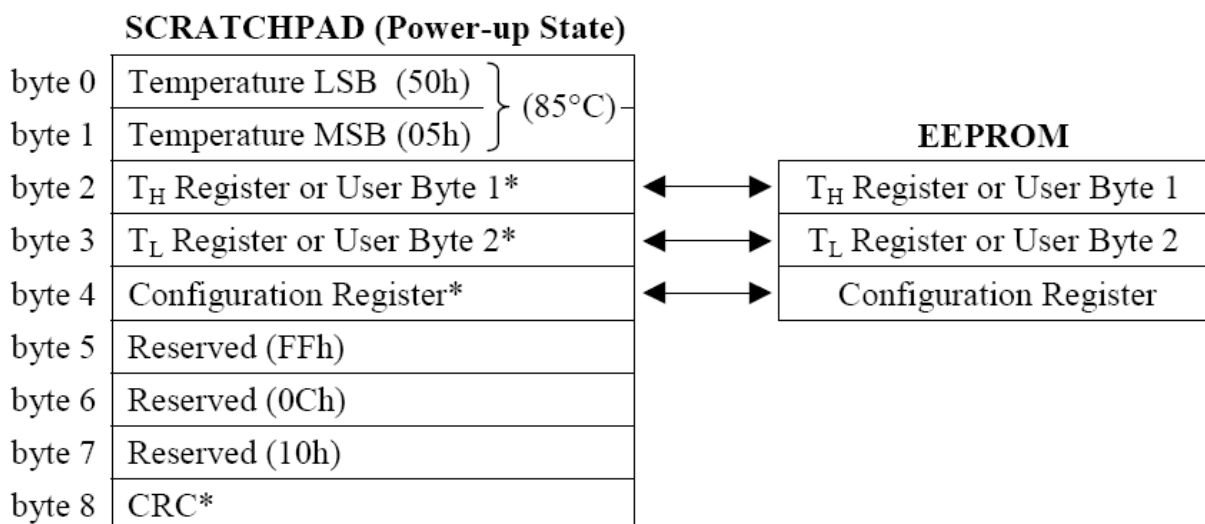
8 Байт ОЗУ только для чтения и содержат циклический контроль избыточности (CRC) код, вычисленный для байтов 0 - 7 ОЗУ. DS18B20 генерирует циклический контроль избыточности, используя метод, описанный в разделе генерация циклического контроля избыточности.

Для записи данных в байты 2, 3, и 4 ОЗУ используем команду Write Scratchpad [4Eh]; данные должны быть переданы к DS18B20, начиная с байта 2. Чтобы проверить корректность записи данных, необходимо выполнить чтение (используя команду чтения Read Scratchpad [BEh]) после того, как данные будут записаны. Обратите внимание, что при чтении данные DS18B20 начинает передавать с 0 байта. Чтобы сохранить T<sub>H</sub>, T<sub>L</sub> и данные регистра конфигурации в EEPROM, устройство управление должно выдать команду Copy Scratchpad [48h].

Данные, сохраненные в регистрах EEPROM, при включении питания перезагружаются в ОЗУ.

Данные могут быть перезагружены из EEPROM в ОЗУ в любое время командой Recall E<sup>2</sup> [B8h]. Устройство управления может контролировать операцию Recall E<sup>2</sup> [B8h] (процесс вызова данных из EEPROM в ОЗУ) путем выдачи синхроимпульса после команды и контроля состояния шины, если 0 операция перезагрузки продолжается, если 1 процесс выполнен.

### DS18B20 MEMORY MAP Figure 7



\*Power-up state depends on value(s) stored in EEPROM



## РЕГИСТР КОНФИГУРАЦИИ

Байт 4 памяти содержит регистр конфигурации, который организован, как иллюстрировано в иллюстрации 8. Пользователь может настроить конверсионную разрешающую способность DS18B20, используя R0 и биты R1 в этом регистре как показано в Таблице 3. Значение по умолчанию включения питания этих битов - R0 = 1 и R1 = 1 (12-битовая разрешающая способность). Обратите внимание, что есть прямая зависимость между разрешающей способностью и конверсионным временем. Бит 7 и биты от 0 до 4 в регистре конфигурации зарезервированы для внутреннего использования устройством и не могут быть изменены пользователем, при чтении эти биты возвращают «1».

### РЕГИСТР КОНФИГУРАЦИИ Figure 8

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0	R1	R0	1	1	1	1	1

### КОНФИГУРАЦИЯ РАЗРЕШАЮЩЕЙ СПОСОБНОСТИ ТЕРМОМЕТРА Table 3

R1	R0	Разрешение	Максимальное время конвертирования	
0	0	9-bit	93.75 ms	(tCONV/8)
0	1	10-bit	187.5 ms	(tCONV/4)
1	0	11-bit	375 ms	(tCONV/2)
1	1	12-bit	750 ms	(tCONV)

## CRC GENERATION

CRC bytes are provided as part of the DS18B20's 64-bit ROM code and in the 9th byte of the scratchpad memory. The ROM code CRC is calculated from the first 56 bits of the ROM code and is contained in the most significant byte of the ROM. The scratchpad CRC is calculated from the data stored in the scratchpad, and therefore it changes when the data in the scratchpad changes. The CRCs provide the bus master with a method of data validation when data is read from the DS18B20. To verify that data has been read correctly, the bus master must re-calculate the CRC from the received data and then compare this value to either the ROM code CRC (for ROM reads) or to the scratchpad CRC (for scratchpad reads). If the calculated CRC matches the read CRC, the data has been received error free. The comparison of CRC values and the decision to continue with an operation are determined entirely by the bus master. There is no circuitry inside the DS18B20 that prevents a command sequence from proceeding if the DS18B20 CRC (ROM or scratchpad) does not match the value generated by the bus master.

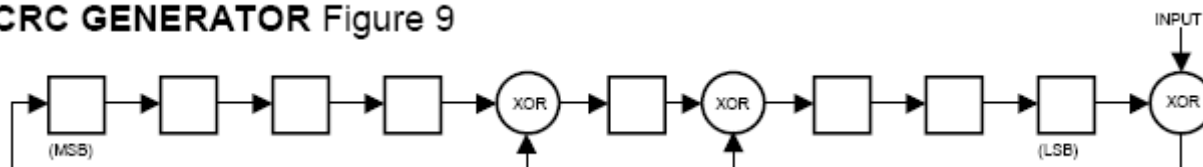
The equivalent polynomial function of the CRC (ROM or scratchpad) is:

$$\text{CRC} = X^8 + X^5 + X^4 + 1$$

The bus master can re-calculate the CRC and compare it to the CRC values from the DS18B20 using the polynomial generator shown in Figure 9. This circuit consists of a shift register and XOR gates, and the shift register bits are initialized to 0. Starting with the least significant bit of the ROM code or the least significant bit of byte 0 in the scratchpad, one bit at a time should be shifted into the shift register. After shifting in the 56th bit from the ROM or the most significant bit of byte 7 from the scratchpad, the polynomial generator will contain the re-calculated CRC. Next, the 8-bit ROM code or scratchpad CRC from the DS18B20 must be shifted into the circuit. At this point, if the re-calculated CRC was correct, the shift register will contain all 0s. Additional information about the Dallas 1-Wire cyclic redundancy check

DS18B20 is available in *Application Note 27: Understanding and Using Cyclic Redundancy Checks with Dallas Semiconductor Touch Memory Products*.

**CRC GENERATOR** Figure 9



## 1-WIRE BUS SYSTEM

The 1-Wire bus system uses a single bus master to control one or more slave devices. The DS18B20 is always a slave. When there is only one slave on the bus, the system is referred to as a "single-drop" system; the system is "multidrop" if there are multiple slaves on the bus.

All data and commands are transmitted least significant bit first over the 1-Wire bus.

The following discussion of the 1-Wire bus system is broken down into three topics: hardware configuration, transaction sequence, and 1-Wire signaling (signal types and timing).

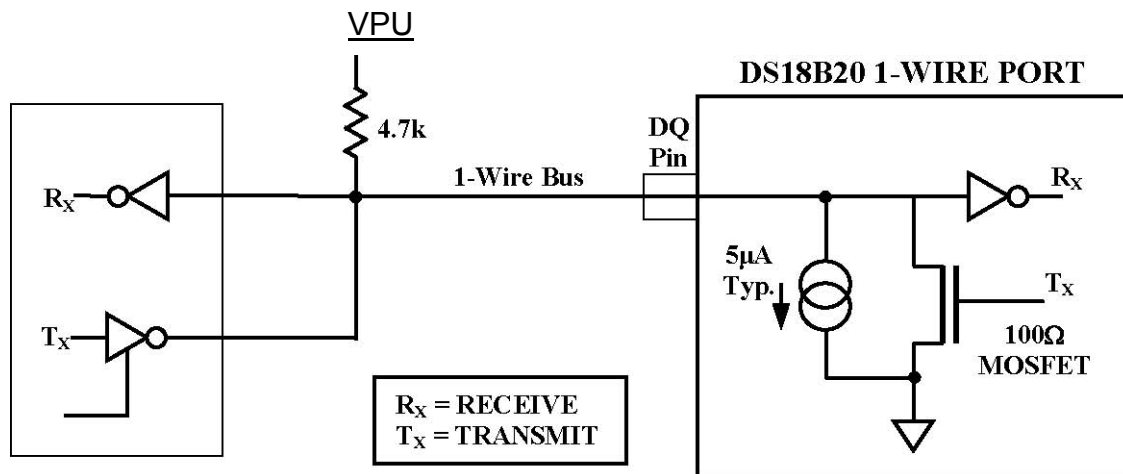
## HARDWARE CONFIGURATION

The 1-Wire bus has by definition only a single data line. Each device (master or slave) interfaces to the data line via an open-drain or 3-state port. This allows each device to "release" the data line when the device is not transmitting data so the bus is available for use by another device. The 1-Wire port of the DS18B20 (the DQ pin) is open drain with an internal circuit equivalent to

that shown in Figure 10.

The 1-Wire bus requires an external pullup resistor of approximately  $5k\Omega$ ; thus, the idle state for the 1Wire bus is high. If for any reason a transaction needs to be suspended, the bus MUST be left in the idle state if the transaction is to resume. Infinite recovery time can occur between bits so long as the 1-Wire bus is in the inactive (high) state during the recovery period. If the bus is held low for more than  $480\mu s$ , all components on the bus will be reset.

#### HARDWARE CONFIGURATION Figure 10



#### TRANSACTION SEQUENCE

##### ПОСЛЕДОВАТЕЛЬНОСТЬ ОПЕРАЦИЙ

Последовательность операций для обращения к DS18B20:

**Шаг 1.** Инициализация

**Шаг 2.** Команда ROM (сопровождается любым требуемым обменом данными)

**Шаг 3.** Функциональная Команда DS18B20 (сопровождается любым требуемым обменом данными)

Очень важно следовать за этой последовательностью каждый раз, когда обращаются к DS18B20, поскольку DS18B20 не будет отвечать, если любые шаги в последовательность отсутствуют или не в порядке.

Исключения из этого правила составляют команды – Поиск ROM [F0h] и Тревога Ищет на командах [ECh].

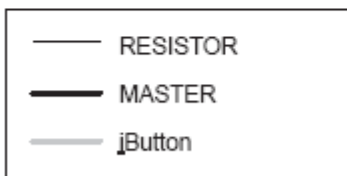
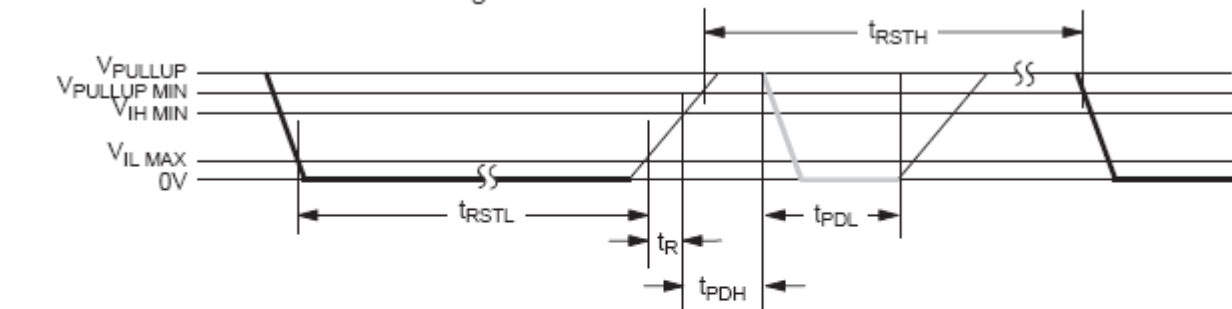
После подачи любой из этих команд, контроллер управления (управляемая программа) должен возвратиться к Шагу 1 в последовательности операций обращения.

## INITIALIZATION

Все операции на шине (1-Wire bus) начинаются с последовательности **инициализации**. Последовательность инициализации состоит из **импульса сброса**, переданного устройством управления шиной, сопровождаемым **импульсом(ами) присутствия**, переданными подчиненными устройствами. Импульс присутствия позволяет устройству управления шиной знать, что подчиненные устройства (типа DS18B20) присутствуют на шине и готовы к работе.

Временные характеристики для сброса и импульсов присутствия детализированы в разделе **ПРОЦЕДУРА ИНИЦИАЛИЗАЦИИ: ИМПУЛЬСЫ СБРОСА И ПРИСУТСТВИЯ**.

### RESET AND PRESENCE PULSE Figure 4-4



#### Regular Speed

$480 \mu\text{s} \leq t_{RSTL} < \infty$  \*  
 $480 \mu\text{s} \leq t_{RSTH} < \infty$  (includes recovery time)  
 $15 \mu\text{s} \leq t_{PDH} < 60 \mu\text{s}$   
 $60 \mu\text{s} \leq t_{PDL} < 240 \mu\text{s}$

#### Overdrive Speed

$48 \mu\text{s} \leq t_{RSTL} < 80 \mu\text{s}$   
 $48 \mu\text{s} \leq t_{RSTH} < \infty$   
 $2 \mu\text{s} \leq t_{PDH} < 6 \mu\text{s}$   
 $8 \mu\text{s} \leq t_{PDL} < 24 \mu\text{s}$

\* In order not to mask interrupt signalling by other devices on the 1-Wire bus,  $t_{RSTL} + t_R$  should always be less than 960  $\mu\text{s}$ .

## ROM COMMANDS

### Команды ROM

После того, как устройство управления шиной обнаружило импульс присутствия, оно может формировать команды ROM. Эти команды оперируют уникальными кодами ROM на 64 бита для каждого подчиненного устройства, и позволяют устройству управления выбирать определенное устройство, из многих устройств присутствующих на шине.

Эти команды также позволяют устройству управления определять как много и какие типы устройств присутствуют на шине, а также определять любое устройство, находящееся в **состоянии Тревога**.

Есть пять команд ROM, и каждая команда 8 битов длиной. Главное устройство должно передать соответствующую команду ROM перед передачей команды функции DS18B20. Блок-схема для операций команд ROM показывают в Рисунке 11.

## SEARCH ROM [F0h] - (ПОИСК ROM)

Когда система первоначально включена, главное устройство должно идентифицировать коды ROM всех подчиненных устройств на шине, эта команда позволяет устройству управления определять номера и типы подчиненных устройств. Устройство управления изучает коды ROM через процесс устранения, которое требует, чтобы Главное устройство исполнил цикл Поиска ROM (то есть, команда ROM Поиска, сопровождаемая обменом данных). Эту процедуру необходимо выполнить столько раз, сколько необходимо, чтобы идентифицировать все из подчиненных устройств. Если есть только одно подчиненное устройство на шине, более простая команда Чтения ROM (см. ниже) может использоваться вместо процесса Поиска ROM.

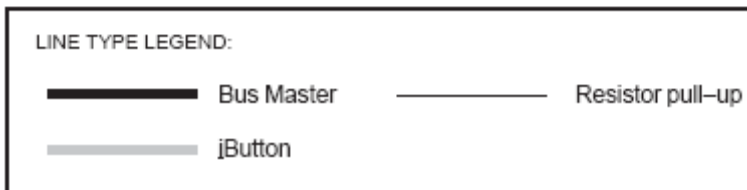
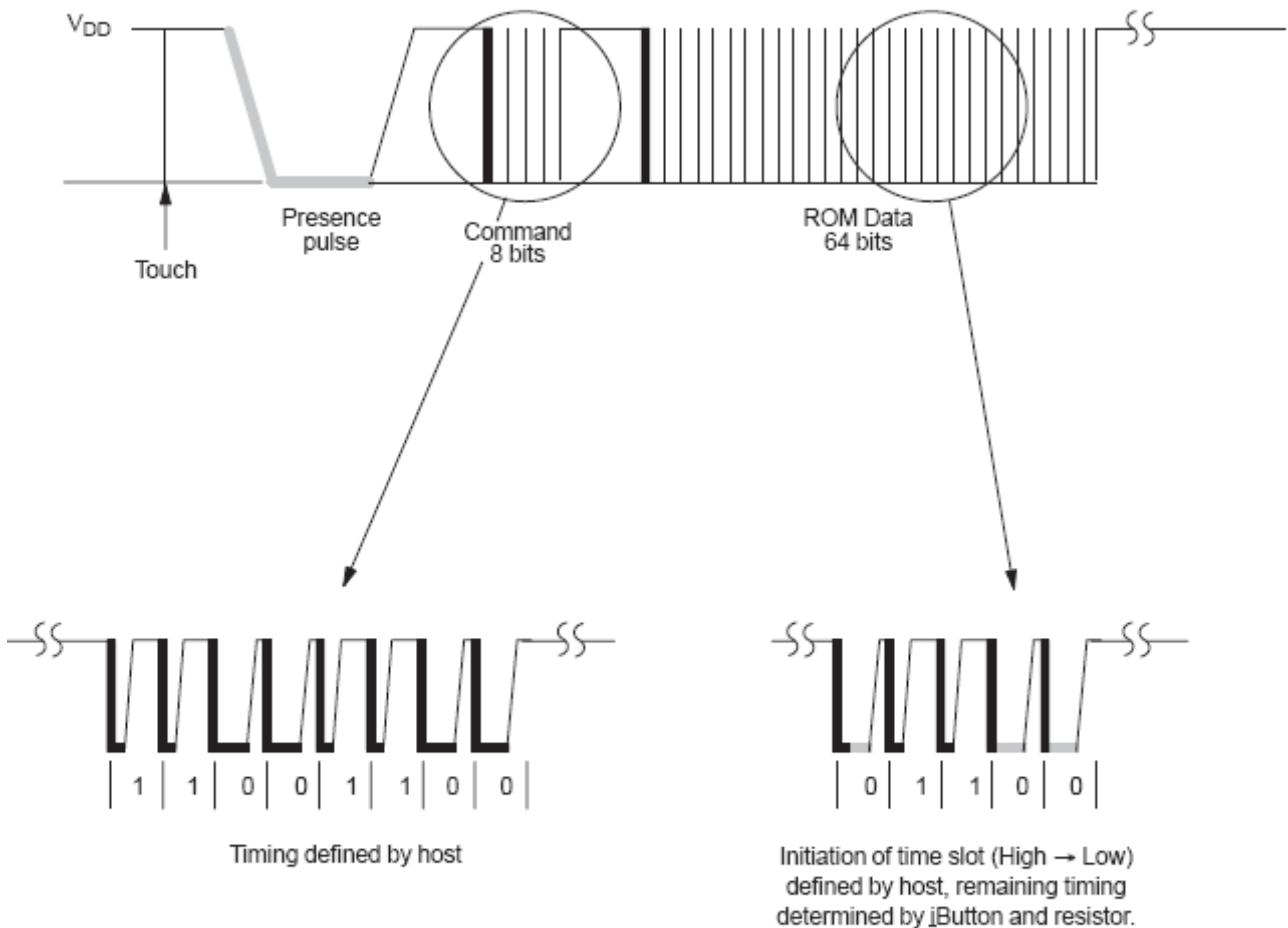
Для детального объяснения процедуры Поиска ROM, обратитесь ® *Книге Стандартов* в [www.ibutton.com/ibuttons/standard.pdf](http://www.ibutton.com/ibuttons/standard.pdf).

После каждого цикла Поиска ROM, устройство управления шиной должно возвратиться к Шагу 1 (Инициализация) в операционной последовательности.

## READ ROM [33h]

Эта команда может только использоваться, когда есть одно подчиненное устройство на шине. Эта команда позволяет устройству управления шиной читать ROM подчиненного устройства (код 64 бита), не используя процедуру Поиска ROM. Если эта команда используется там, где больше чем одно подчиненное устройство на шине, произойдет конфликт на уровне данных, так как все подчиненные устройства сделают попытку ответить в одно и то же время.

**EXAMPLE READ ROM** Figure 1-7



### **MATCH ROM [55h] Соответствие ROM [55h]**

Команда соответствия ROM, сопровождаемая последовательностью кода ROM на 64 бита позволяет устройству управления шиной обращаться к определенному подчиненному устройству на шине. Только подчиненное устройство, которое точно соответствует 64 битам последовательности кода ROM, ответит на функциональную команду, выпущенную главным устройством. Все другие подчиненные устройства на шине будут ждать импульса сброса.

### **SKIP ROM [CCh] Пропуск ROM [CCh]**

Главное устройство может использовать эту команду, чтобы обратиться ко всем устройствам на шине одновременно. Например, главное устройство может заставить чтобы все DS18B20 (датчики температуры) на шине, начали одновременно температурные преобразования. Для этого необходимо выдать на шину команду Пропуска ROM [CCh] сопровождаемую командой Температурного преобразования [44h].

*Обратите внимание*, что команда ЧТЕНИЕ ПАМЯТИ [BEh] может следовать за командой Пропуска ROM, только если на шине присутствует одно подчиненное устройство. Команда Пропуска ROM, сопровождаемая командой ЧТЕНИЕ ПАМЯТИ вызовет конфликт на уровне данных на шине, если на шине более одного подчиненного устройства, так как все устройства будут пытаться одновременно передавать данные.

### **ALARM SEARCH [ECh] Поиск Тревоги [ECh] (состояние Тревога)**

Операция этой команды идентична операции команды Поиска ROM за исключением того, что только DS18B20 (датчики температуры) с установленным флажком аварии ответят. Эта команда позволяет главному устройству определять, какие DS18B20-сы испытали сигнальное состояние в течение недавнего температурного преобразования. После каждого цикла Поиска **Тревоги** (то есть, команда Alarm Search, сопровождаемая обменом данными), устройство управления шиной должно возвратиться к Шагу 1 (Инициализация) в операционной последовательности.

## **ФУНКЦИОНАЛЬНЫЕ КОМАНДЫ - DS18B20**

После того, как устройство управления шиной обработало команду ROM, чтобы обратиться к DS18B20, с которым оно желает связаться, устройство управления может формировать одну из команд функции DS18B20. Эти команды позволяют выполнить функции записи или чтения оперативной памяти DS18B20, инициализировать температурные преобразования или определить режим электропитания.

Функциональные команды DS18B20 описанные ниже, сведены в Таблицу 4 и проиллюстрированы на блок-схеме, Рисунок 12.

### **Конвертировать температуру [44h]**

Эта команда начинает единственное температурное преобразование. После окончания преобразования данные сохраняются в 2-байтовом температурном регистре в оперативной памяти, а DS18B20 **возвращается в неактивное состояние** с низким энергопотреблением. Если устройство используется в режиме паразитного питания, то в пределах не позже 10 мкс (максимальный) после подачи команды устройство управления должно установить высокий уровень на шине на время продолжительности преобразование (tconv).

Если DS18B20 **питается от внешнего источника питания**, главное устройство может считывать состояние шины после команды Конвертирования температуры [44h]. Если на шине логический «Ноль» - это значит, что DS18B20 выполняет температурное преобразование. Если на шине логическая «Единица» – это значит, что преобразование окончено и можно считывать данные.

В режиме использования паразитного питания эта методика уведомления не может использоваться, так как шина должна быть в высоком состоянии в течение всего цикла преобразования температуры.

### **Запись в память [4Eh]**

Эта команда позволяет устройству управления записывать 3 байта данных в память DS18B20.

Первый байт данных записывается в регистр (ТН), второй байт записывается в регистр (ТL), и третий байт записывается в регистр конфигурации.

Данные должны быть переданы наименьшим значащим битом вперед.

Для датчиков температуры с паразитным питанием все три байта ДОЛЖНЫ быть записаны командой КОПИРОВАНИЕ ОЗУ В ПЗУ [48h] прежде, чем устройством управления будет сгенерирован импульс сброса, иначе данные будут потеряны.

### **Чтение памяти [BEh]**

Эта команда позволяет хозяину читать содержание ПАМЯТИ. Передача данных начинается с наименьшего значащего бита байта 0 и продолжается до 9-ого байта (байт 8 - циклический контроль избыточности). Устройство управления может выполнить сброс, чтобы закончить чтение в любое время, если необходимо только часть данных.

### **Копирование ОЗУ В ПЗУ [48h]**

Эта команда копирует содержание регистров (ТН, ТL) и регистра конфигурации (байты 2, 3 и 4) в ПЗУ.

Если устройство используется в режиме паразитного питания, то в пределах не позднее 10  $\mu$ s (максимально) после подачи этой команды устройство управления должно установить высокий уровень на шине и поддерживать его в течении не менее 10ms.

### **Повторная загрузка [B8h]**

Эта команда повторно загружает значения регистров (ТН, ТL) и данные в регистр конфигурации с ПЗУ и размещает данные в байты 2, 3, и 4, соответственно, в памяти. Главное устройство может контролировать процесс загрузки ОЗУ из ПЗУ считывая состояние шины после команды ПОВТОРНАЯ ЗАГРУЗКА. Если на шине логический «ноль» - это значит идет операция перезагрузки, если логическая «1» Операция выполнена.

Операция ПОВТОРНАЯ ЗАГРУЗКА [B8h] выполняется автоматически при включении питания, данные доступны сразу после включения питания.

### **Вид электропитания датчика [B4h]**

Главное устройство генерирует эту команду, чтобы определить используют DS18B20s на шине паразитное питание. Если после подачи команды на шине присутствует логический «ноль» - это значит, что DS18B20 использует паразитное питание. Иначе DS18B20 использует внешнее питание (логическая единица).



## С.2. Краткий обзор Команды

Устройства iButtons могут работать как автономно на шине, так и поддерживают следующие Сетевые Команды на основе ROM:

- Чтение ROM                   **[33h]**            (*Read*)
- Пропуск ROM               **[CCh]**            (*Skip*)
- Соответствие ROM       **[55h]**            (*Match*)
- Поиск ROM                 **[F0h]**            (*Search*)

После выполнения любой команды ROM, устройства достигает уровня Транспорта (способно передавать свои данные памяти).

**Чтение ROM [33h] (*Read*)** используется, чтобы прочитать ROM память устройства, если на шине оно только одно. После отправки этой команды Главное устройство должен генерировать 64 слота времени считывания. iButton пошлет содержимое его ROM памяти младшим битом вперед, начиная с кода семейства, сопровождаемого серийным номером и байтом циклического контроля избыточности.

Если на шине несколько iButtons устройств, то для чтения ROM памяти необходимо воспользоваться командой **Поиск ROM [F0h] (*Search*)**, чтобы определить содержание ROM памяти устройств прежде, чем к ним можно будет обратиться.

Если содержание ROM памяти не представляет интерес, потому что на шине только одно iButton, поиск может быть пропущен, посылая команду **Пропуск ROM [CCh] (*Skip*)**. Немедленно после этой команды, устройство достигает Транспортного уровня.

Команды **Соответствие ROM [55h] (*Match*)** может использоваться, чтобы обратиться к интересующему устройству, если на шине присутствует несколько iButtons устройств.

Код ROM выполняет функцию адреса устройства, чтобы активизировать его Транспортный уровень. Тот же самый Код ROM не может активировать более 1 устройства, так как соответствие кодов ROM только одному устройству определено при их производстве. Если два iButtons имеют то же самое серийный номер, их семейные коды будут отличны. Этим способом, исключается любой беспорядок или неопределенность.

После подачи команды **Соответствие ROM [55h] (*Match*)**, Главное устройством будет послано в течение следующих 64 слотов времени содержание ROM памяти требуемого устройства. Последовательность битов должна быть тем же самым, как они были получены при чтении ROM, то есть, младшим битом вперед, начинаясь с семейного кода, сопровождаемого серийным номером и циклическим контролем избыточности. Все iButtons, ROM которого не соответствует требуемому коду, останутся в неактивном состоянии пока они получают другой Импульс Сброса.

### С.3. Команда Поиск ROM [F0h] (Search)

Если Главное устройство не знает серийный номер устройства подключенного к шине, то существует возможность идентифицировать коды ROM каждого устройства подключенного к шине. Для этого необходимо использовать команду Поиск ROM [F0h] (Search). Эта команда действует как команда Чтения ROM объединенная с командой Соответствия ROM.

Процесс выглядит следующим образом: После формирования главным устройством команды **Поиск ROM [F0h] (Search)** все устройства iButtons последовательно будут формировать на шине состояние «0» и «1» соответствующие их значению фактического бита ROM в течение **двух Времен** (тактов) считывания после формирования команды ROM Поиска.

Если все устройства содержат в этой позиции двоичного разряда:

- «0», чтение будет «01»;
- «1», результат будет «10»;

Если устройства содержат в этой позиции двоичного разряда и «1» и «0», чтение приведет к «00» битов, указывая на конфликт.

Главное устройство в следующем (третьем такте) слоте Времени формирует разрядное значение 1 или 0, чтобы отобрать устройства, которые останутся в процессе выбора.

Все устройства у которых бит не соответствует биту сформированному главным устройством перейдут в состояние ожидания и будут находиться в нем пока они не получают Импульс Сброса. После первой стадии выбора, будут следовать 63 читающих/выбора цикла, пока, наконец, главное устройство не определит Код ROM одного подчиненного устройства и обратиться к нему.

Каждая стадия выбора состоит из двух слотов Времени считывания и один слот Времени записи. Полный процесс изучения и одновременная адресация - приблизительно три раза длина команды ROM Соответствия, но это позволяет выбрать из всех связанных устройств последовательно все коды ROM.

В приложении, где iButtons устройства подключены к одной шине, это является самым эффективным способом, чтобы определить коды все ROM подчиненных устройств. После чего главное устройство может использовать команду ROM Соответствия, чтобы обратиться к определенному устройству.

Если приложение требует постоянной идентификации и коммуникации с новыми устройствами, так как они могут подключаться и отключаться в ходе работы, то устройство управления должно будет использовать команду Поиск ROM, чтобы идентифицировать коды ROM для обращаться к каждому новому устройству.

Блок-схеме всех Команд ROM показывают в иллюстрации 5-2.

Так как логика команды ROM Поиска самый сложный процесс, следующий пример используется, чтобы иллюстрировать это шаг за шагом.

Четыре устройства установлены на шине. Их двоичное содержание ROM следующее:

устройство 1: xxxxxx10101100  
устройство 2: xxxxxx01010101  
устройство 3: xxxxxx10101111  
устройство 4: xxxxxx10001000

для упрощения символом «x» заменены старшие биты и показаны только младшие восемь битов содержания ROM. Поиск младшего бита происходит следующим образом:

1. Главное устройство начинает последовательность инициализации
  - формирует **Импульс Сброса**.
  - iButtons отвечают формированием **импульсов Присутствия**.

2. Тогда Главное устройство формирует команду **Поиск ROM**.

3. Главное устройство читает один бит с шины. Каждое устройство ответит, помещая значение первого бита соответствующего его данным ROM. Устройства 1 и 4 поместят «0» на шину, то есть, они установят на шине низкий уровень. Устройства 2 и 3 сформируют «1» позволяя на линии оставаться в высоком уровне. Результат – «логическое И» всех устройств на линии; поэтому Главное устройство читает 0.

Главное устройство будет читать следующий бит. (С тех пор когда команда Search ROM выполняется, все устройства отвечают одновременно). Все устройства помещают на шину **дополнение первого бита** их соответствующего. Устройства 1 и 4 сформируют «1»; устройства 2 и 3 сформируют «0». Таким образом, на шине будет состояние логического «0». Главное устройство снова читает «0» при формировании дополнительного кода первого информационного разряда ROM (чтение дает «00» - *состояние разрядных конфликтов*). Это говорит Главному устройству, что есть устройства на шине содержащие в первом бите как «0», так и «1».

Если бы все устройства имели «0» в этой позиции двоичного разряда, чтение дало бы результат «01»; если бы позиция двоичного разряда содержала во всех устройства «1» результат был бы «10».

4. Главное устройство теперь решает писать «0» и формирует запись его на шину. Эта операция переводит Устройства 2 и 3 (содержащие в этом разряде «1») в пассивное состояние, оставляя только устройства 1 и 4 для участия в процессе поиска.

5. Главное устройство выполняет еще два чтения и получает «01». Это говорит, что все активные устройства имеют 0 в этой позиции двоичного разряда их ROM.

6. Главное устройство тогда пишет 0, чтобы сохранить устройства 1 и 4 активными.

7. Главное устройство выполняет два чтения и получает два «00» биты. Это снова указывает, что в этом разряде присутствуют устройства имеющие «1» и «0».

8. Главное устройство снова пишет 0. Это деактивирует устройство 1, оставляя устройство 4 как единственный активный элемент.

9. Следующие чтения до конца ROM не будут давать состояние разрядных конфликтов. Отсутствие разрядных конфликтов до конца цикла поиска говорит, что происходит чтение ROM только одного активного элемента. Прочитав следующий бит Главное устройство снова посылает этот бит, чтобы сохранить устройство активным. Как только все биты ROM устройства известны и последний бит снова послан Главным устройством, устройство готово к принятию команду Транспортного уровня (для обмена информацией).

- 10. Главное устройство должно изучить данные ROM других устройств.**

Поэтому оно запускает следующую последовательность Поиска ROM, повторяя шаги 1 - 7.

11. **В самой старшей позиции** двоичного разряда, где Главное устройство писало «0» в первом проходе (шаг 8), оно теперь пишет «1». Это снимает выделение устройства 4, оставляя устройство 1 активным.

12. Как в шаге 9, следующие чтения до конца ROM не будут давать состояние разрядным конфликтам. Этим заканчивается второй Поиск ROM, где Главное устройство считывает содержание ROM другого устройства.

- 13. Главное устройство должен изучить данные ROM других устройств.**

Поэтому, оно запускает следующую последовательность Поиска ROM, повторяя шаги 1 - 3.

14. Во втором проходе в наивысшей степени позиция двоичного разряда, где Главное устройство написал 0 в первом проходе (шаг 4), это теперь пишет 1. Это снимает выделение устройств 1 и 4, оставляя устройства 2 и 3 активными.

15. Главное устройство посылает два слота времени считывания и получает два 0 битов, указывая маленький конфликт.

16. Главное устройство снова решает писать 0. Это снимает выделение устройство 3, оставляя устройство 2 как единственное активное устройство.

17. Как в шаге 9, следующие чтения до конца ROM не будут показывать разрядным конфликтам. Этим заканчивается третий Поиск ROM проходит, где Главное устройство имеет изученный содержание другого ROM.

18. Главное устройство должен изучить данные ROM других устройств.

Поэтому это запускает другую последовательность Поиска ROM повторяя шаги 13 - 15.

19. В самой высокой позиции двоичного разряда, где Главное устройство написал а 0 в предыдущем проходе (ступают 16), это теперь пишет 1.

Это снимает выделение устройства 2, оставляя устройство 3 активный.

20. Как в шаге 17, следующие чтения до конца

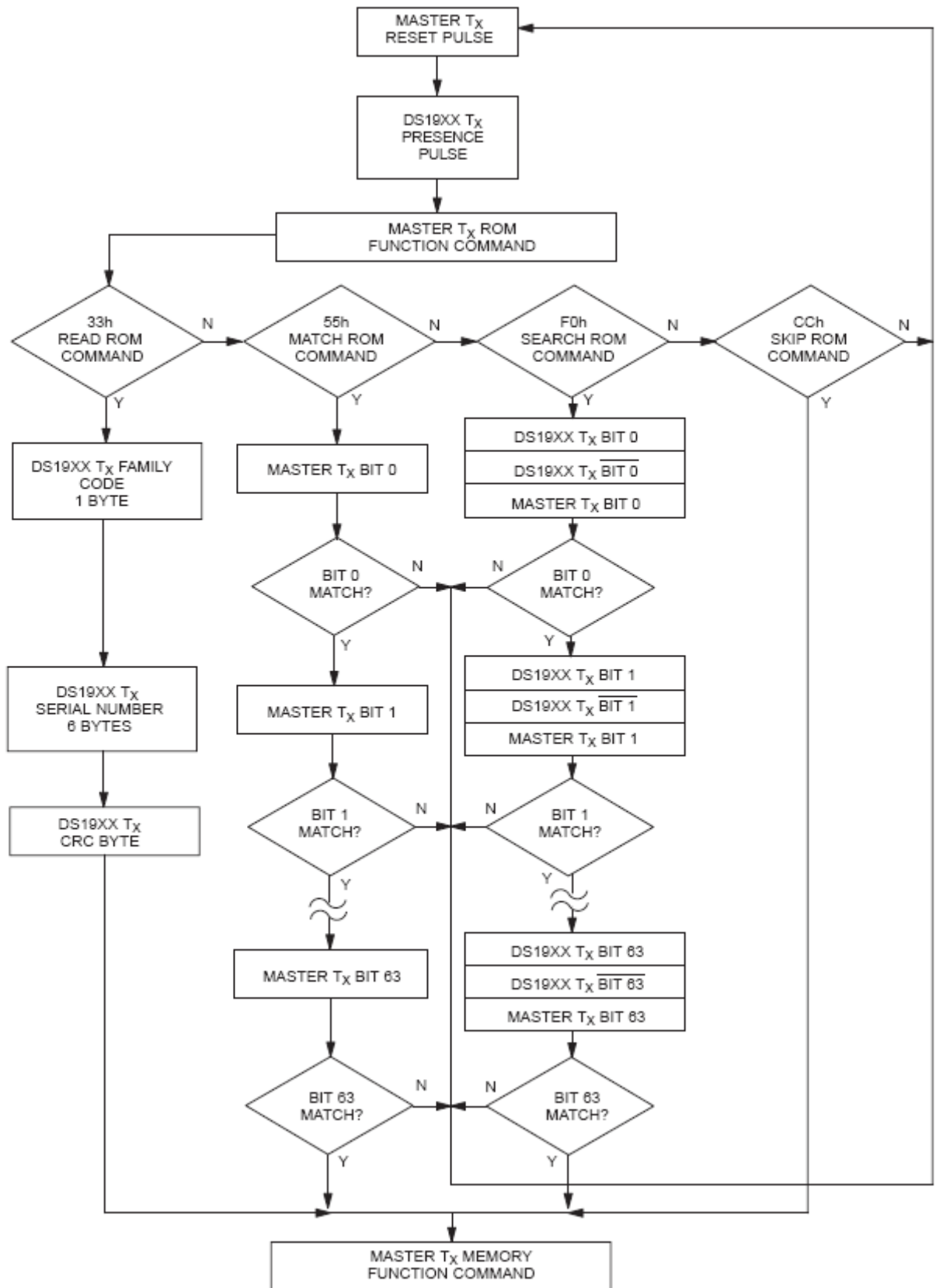
ROM не будут показывать разрядным конфликтам. Это заканчивает четвертый Поиск ROM проходит, где Главное устройство имеет изученный содержание другого ROM.

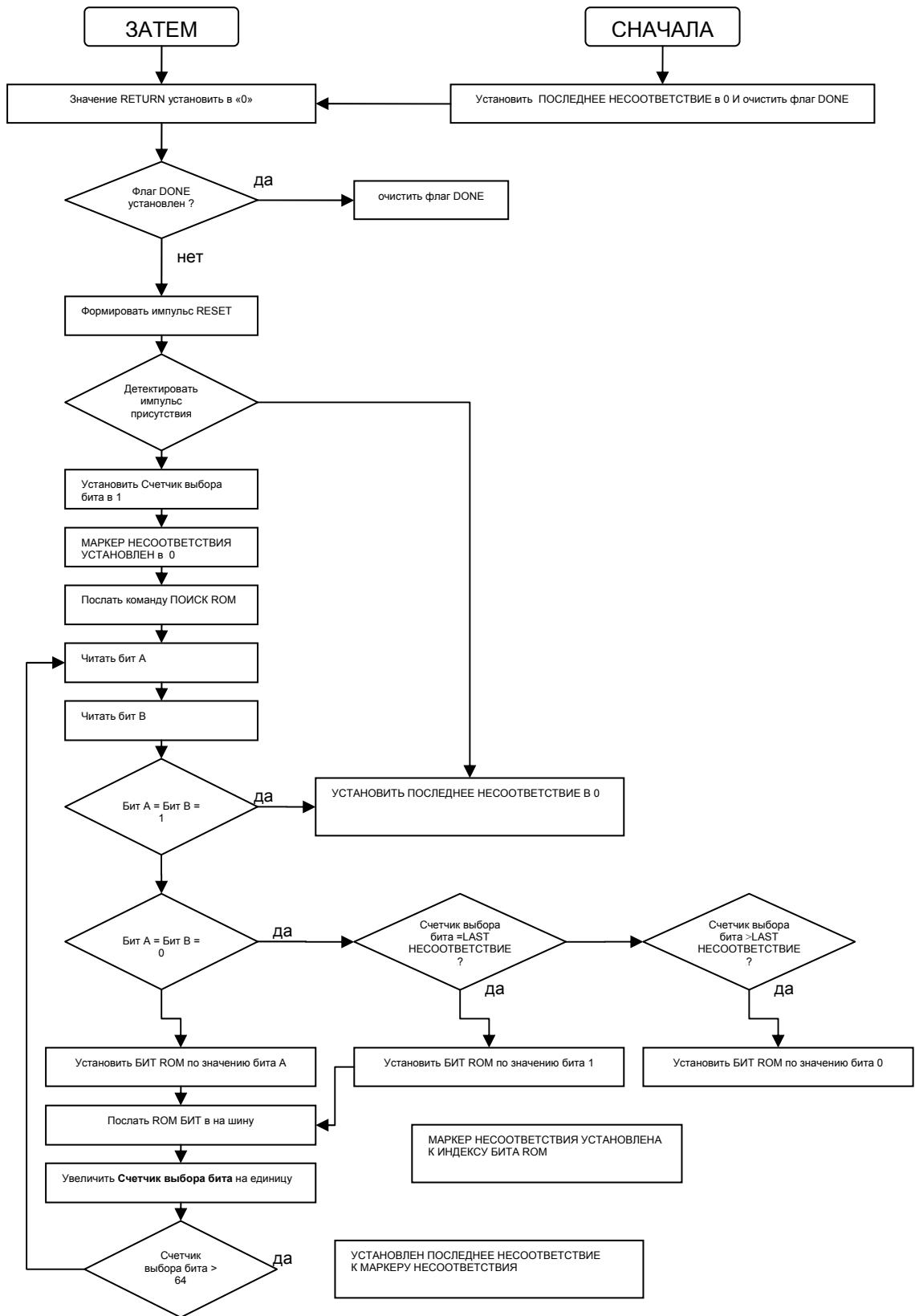
Общий принцип этого процесса поиска должен снять выделение одно устройство за другим в каждой противоречивой позиции двоичного разряда. В конце каждого процесса Поиска ROM, Главное устройство узнал содержание другого ROM. Следующий проход является тем же самым как предыдущий проход до пункта последнее решение. В этом пункте Главное устройство входит в противоположность руководство и продолжается. Если другой конфликт найден, снова 0 написан, и так далее. После обоих путей в самом высоком противоречивая позиция двоичного разряда сопровождается до конца, Главное устройство идет тот же самый путь как прежде, но решающий противоположно в более низкой противоречивой позиции двоичного разряда, и так далее, до всех Данные ROM идентифицированы.

Оптимизированная блок-схема алгоритма ROM Поиска

показанный в иллюстрации 5-3. Это число объясняет, как выполнить общий поиск ROM. Ради этого блок-схема, данные ROM накоплены в маленький массив названный Бит ROM, с битами нумеровал 1 - 64. Установка должна называйте перед любой другой функцией, чтобы инициализировать Система с 1 проводом. Звонок "в Первые" сбросы поиск к начало и идентифицирует первый код ROM, и призывает "Затем" идентифицируйте последовательные коды ROM. Ложное значение возвращенный указывает не больше кодов ROM, которые будут найдены. Время, требуемое изучать содержание одного ROM (нет считая процессорное время Главного устройства) -  $960 \text{ ms} + (8+3*64) * 61 \text{ ms} = 13.16 \text{ ms}$ . Таким образом возможно идентифицировать до 75 различных iButtons в секунду.

ROM FUNCTIONS FLOW CHART Figure 5-2





DS18B20 FUNCTION COMMAND SET Table 4

Команда	Описание	Код	Состояние шины после того, когда передачи команды	Примечания
<b>КОМАНДА КОНВЕРТИРОВАНИЯ ТЕМПЕРАТУРЫ</b>				
Измерение температуры Convert T	Инициализирует начало измерения температуры	44h	DS18B20 выдает на шину результат преобразования температуры.	1
<b>КОМАНДЫ РАБОТЫ С ПАМЯТЬЮ</b>				
Чтение памяти Read Scratchpad	Читает всю память, включая байт циклического контроля избыточности.	BEh	DS18B20 передает до 9 байт данных.	2
Запись в память Write Scratchpad	Записывает три байта в регистры TH, TL и регистр конфигурации.	4Eh	Устройство управления передает 3 байта данных к DS18B20.	3
Копирование ОЗУ в ПЗУ Copy Scratchpad	Копирует значение TH, TL и регистра конфигурации с ОЗУ в ПЗУ.	48h	Ни один	1
Повторная загрузка Recall E2	Повторная загрузка TH, TL и регистра конфигурации с ПЗУ в ОЗУ.	B8h	DS18B20 передает бит состояния.	
Вид электропитания Read Power Supply	Определяет тип питания DS18B20.	B4h	DS18B20 передает бит состояния.	

**ПРИМЕЧАНИЯ:**

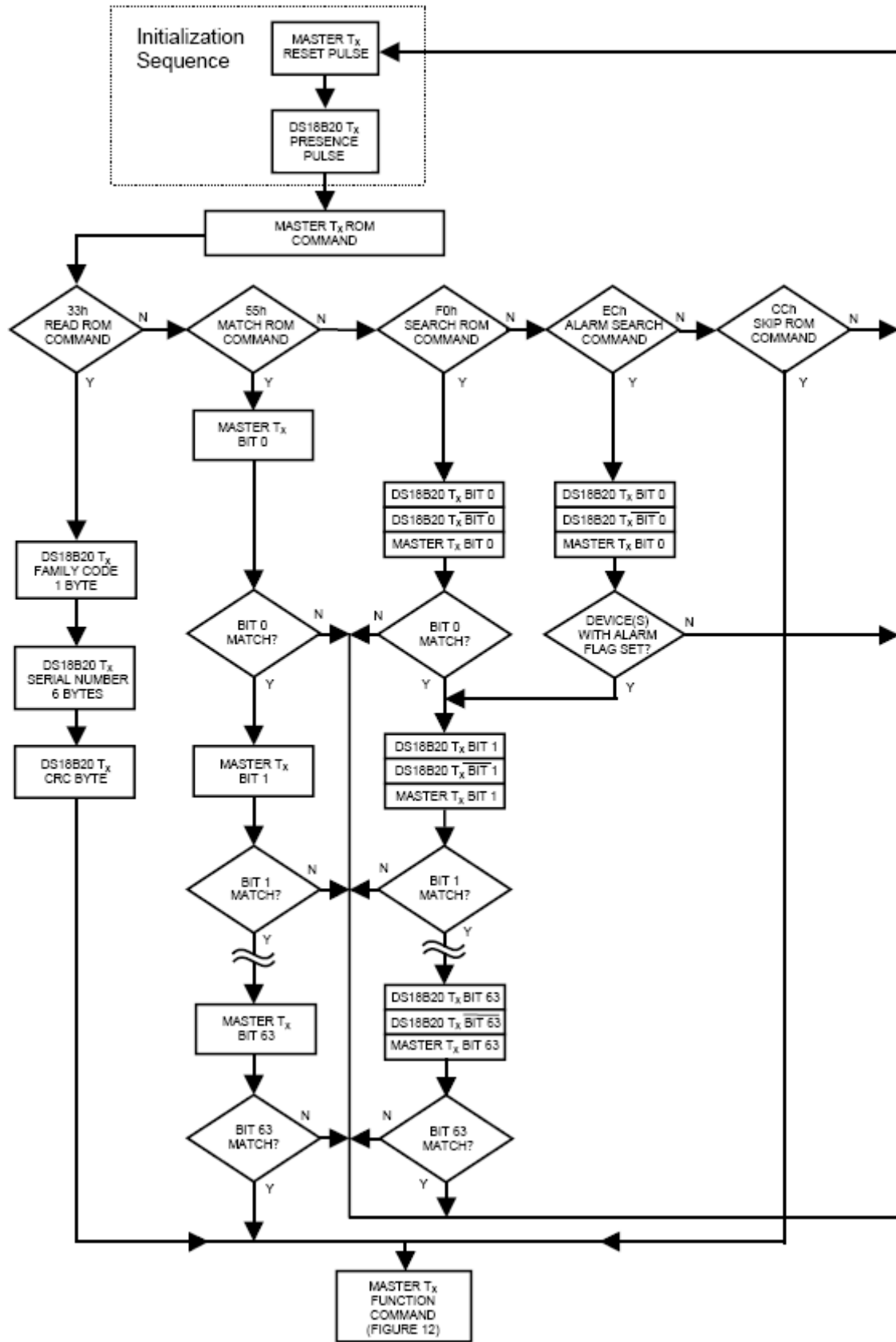
1) При паразитном питании DS18B20s, Главное устройство должно выдать на шину «сильную» 1 в течение температурного преобразования и копирования ОЗУ в ПЗУ. Никакая другая операция на шине в это время не может выполняться.

2) Главное устройство может прервать передачу данных в любое время, импульсом сброс.

3) Все три байта должны быть сохранены в ПЗУ прежде, чем будет сброс.

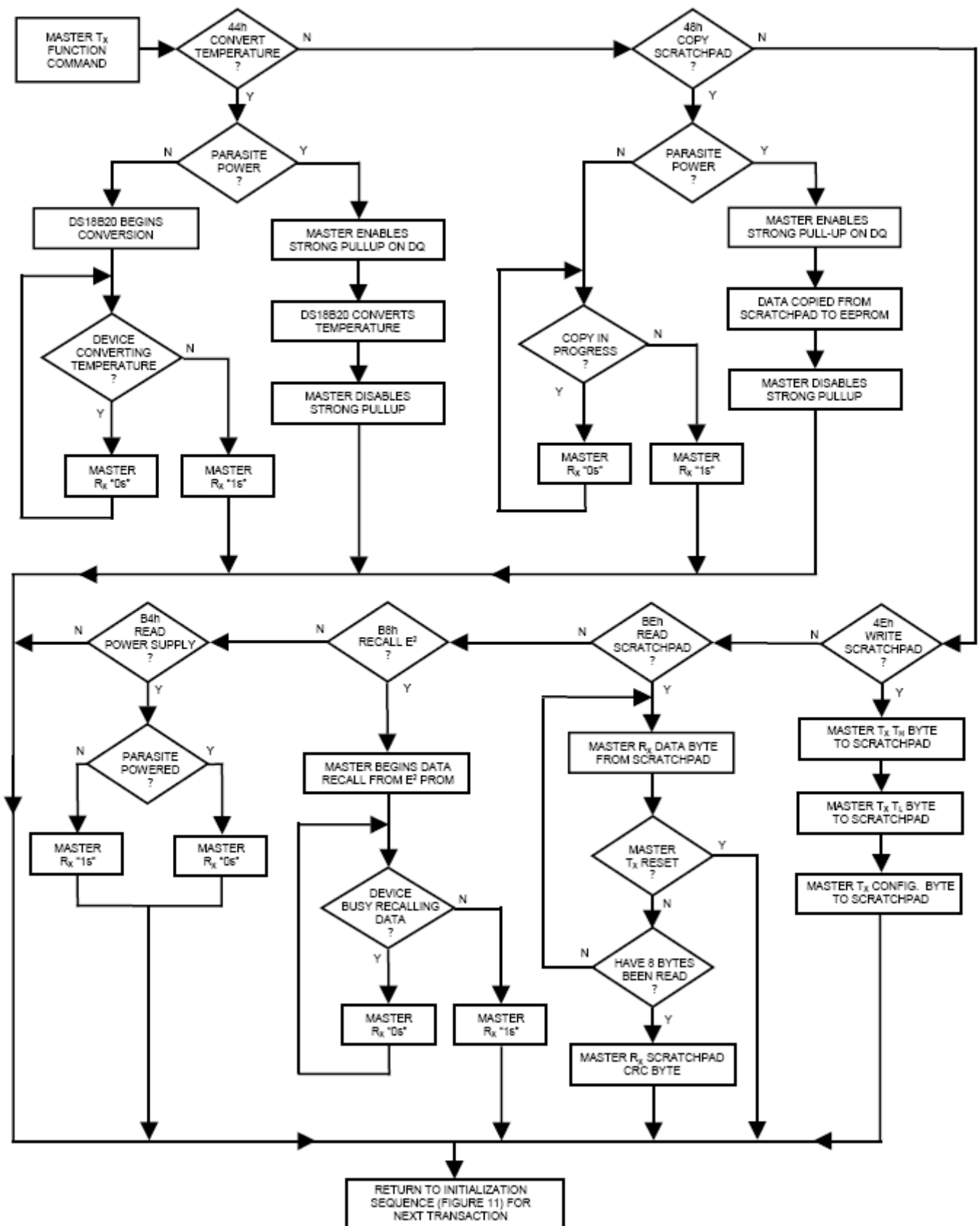
При использовании резистора 1 кОм длина кабеля 200 метров. При чтении температуры до команды температурного преобразования читается число 85.

ROM COMMANDS FLOW CHART Figure 11





# DS18B20 FUNCTION COMMANDS FLOW CHART Figure 12



## 1-WIRE SIGNALING

The DS18B20 uses a strict 1-Wire communication protocol to insure data integrity. Several signal types are defined by this protocol: reset pulse, presence pulse, write 0, write 1, read 0, and read 1. The bus master initiates all of these signals, with the exception of the presence pulse.

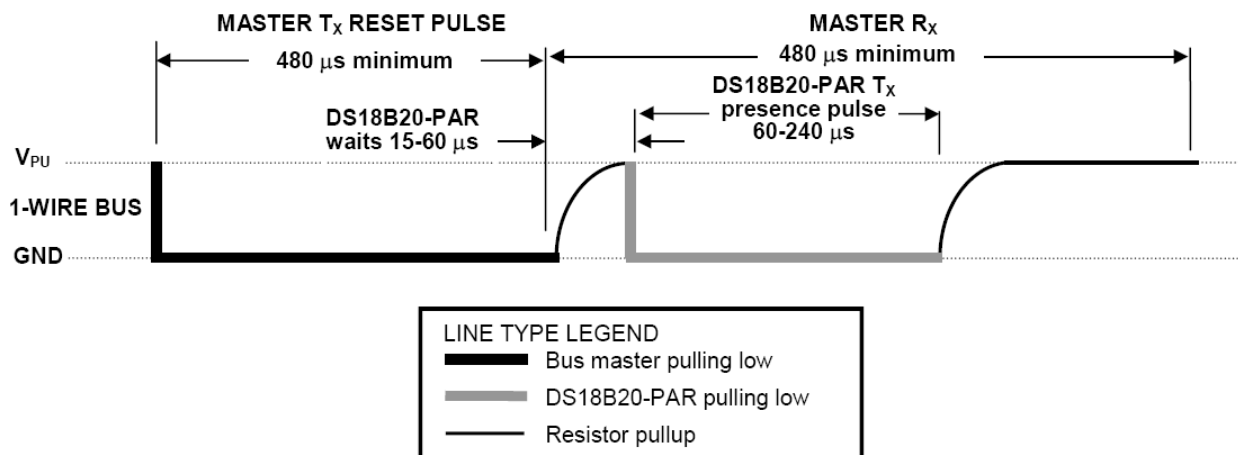
### ПРОЦЕДУРА ИНИЦИАЛИЗАЦИИ: ИМПУЛЬСЫ СБРОСА И ПРИСУТСТВИЯ

#### Процедура инициализации: Сброс и импульс присутствия.

Весь процесс связи с DS18B20 начинается с последовательности инициализации, которая состоит из импульса сброса от устройства управления (УУ), сопровождаемого импульсом присутствия от DS18B20. Это иллюстрировано на рис. 12. Когда DS18B20 посылает импульс присутствия в ответ на сброс, это указывает УУ, что DS18B20 находится на шине и готов работать.

В течение последовательности инициализации устройство управления шиной передает (TX) **импульс сброса**, перемещая шину 1-Wire bus в состояние логического «0» минимум **480  $\mu$ s**. Устройство управления шиной отпускает шину и переходит в режим приема (RX). Когда шина отпущена, (5 кОм max) pullup резистор перемещают шину в уровень логической «1». Когда DS18B20 обнаруживает положительный перепад, он ждет от 15  $\mu$ s до 60 $\mu$ s и затем передает **импульс присутствия**, перемещая шину в логический «0» на длительность от 60  $\mu$ s до 240 $\mu$ s.

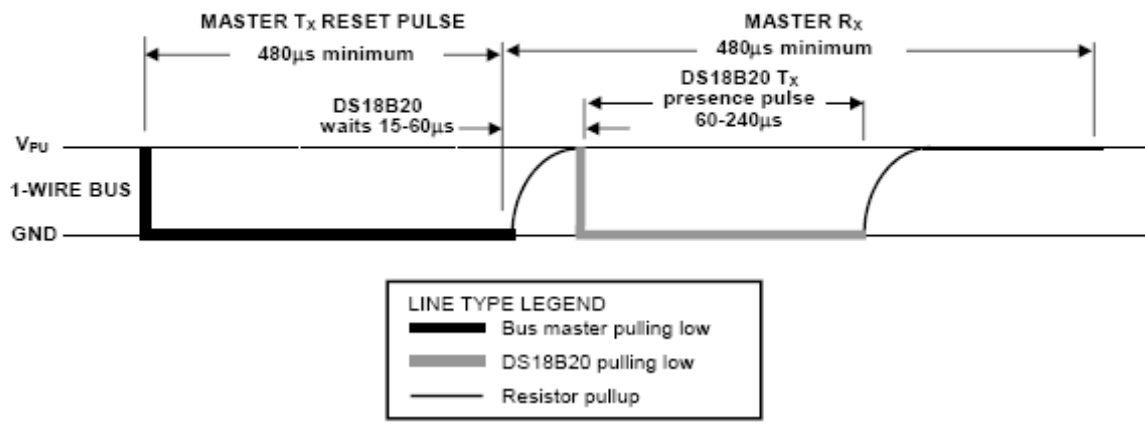
### INITIALIZATION TIMING Figure 12



### ВРЕМЕННЫЕ СЛОТЫ - ЧТЕНИЯ/ЗАПИСИ

Устройство управления шиной пишет данные DS18B20 в течение слотов времени записи и читает данные от DS18B20 в течение слотов времени считывания. Один бит данных передается за один слот времени.

# INITIALIZATION TIMING Figure 13



## СЛОТЫ ВРЕМЕНИ ЗАПИСИ

Есть два типа слотов времени:

- Слот времени записи «1»
- Слот времени записи «0».

Устройство управления шиной использует Слот времени записи «1», чтобы записать логический «1» в DS18B20 и Слот времени записи «0», чтобы записать логический «0» в DS18B20.

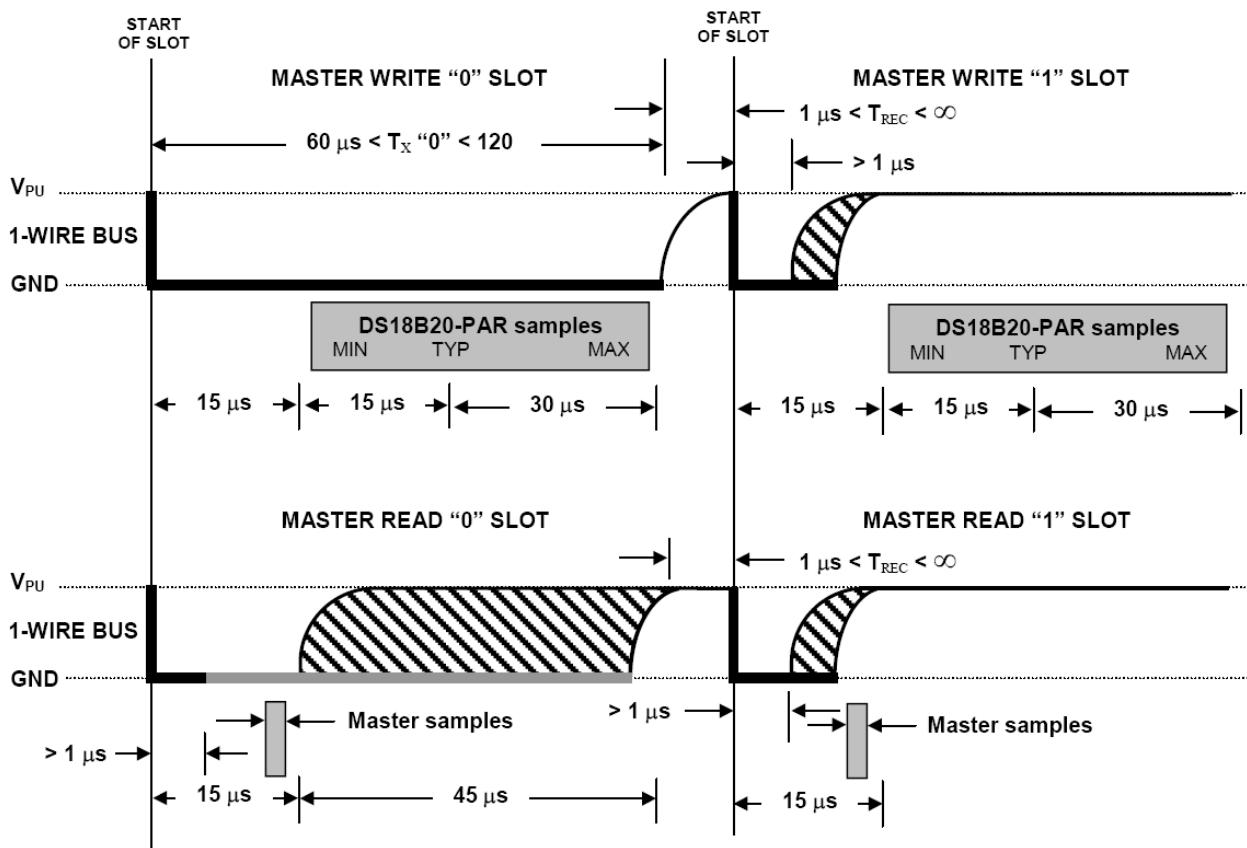
Все слоты времени записи должны быть продолжительностью минимумом 60µs разделенные импульсом восстановления минимумом 1µs. Оба типа слотов времени записи инициализируются устройством управления, устанавливающим на шине логический ноль (см. Рисунок 13).

Чтобы генерировать Слот времени записи «1», после формирования импульса восстановления, устройство управления шиной должно отпустить шину в пределах 15µs. Когда шина отпущена, pullup резистор переместит уровень на шине к логической «1».

Чтобы генерировать Слот времени записи «0», после формирования импульса восстановления, устройство управления шиной должно продолжить удерживать шину продолжительностью всего слота времени (не менее 60µs).

DS18B20 после формирования импульса восстановления выполняет выборку сигнала через 15µs в течение окна, которое продолжается от 15µs до 60µs, и инициализирует слот времени записи. Если уровень на шине высокий в течение окна выборки, осуществляется запись 1 в DS18B20. Если уровень низкий, осуществляется запись 0 в DS18B20.

## READ/WRITE TIME SLOT TIMING DIAGRAM Figure 13



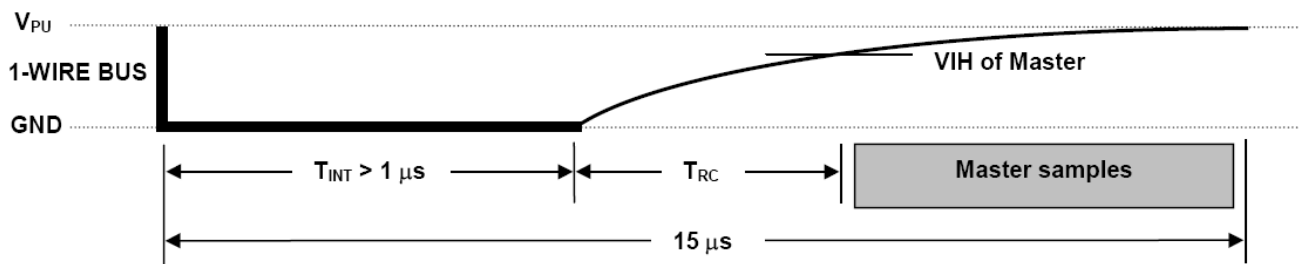
## СЛОТЫ ВРЕМЕНИ СЧИТЫВАНИЯ

DS18B20 может передать данные устройству управления только, когда устройство управления формирует слоты времени считывания. Поэтому, устройство управления должно генерировать слоты времени считывания немедленно после формирования команды Чтения Памяти [BEh] или команды Чтения Вида Питания [B4h], так, чтобы DS18B20 мог обеспечить требуемые данные. Кроме того, устройство управления может генерировать слоты времени считывания после команды Конвертирования [44-ого] или команды Recall E2 [B8h], чтобы узнать состояние операции, как объяснено в разделе КОМАНДЫ ФУНКЦИИ DS18B20

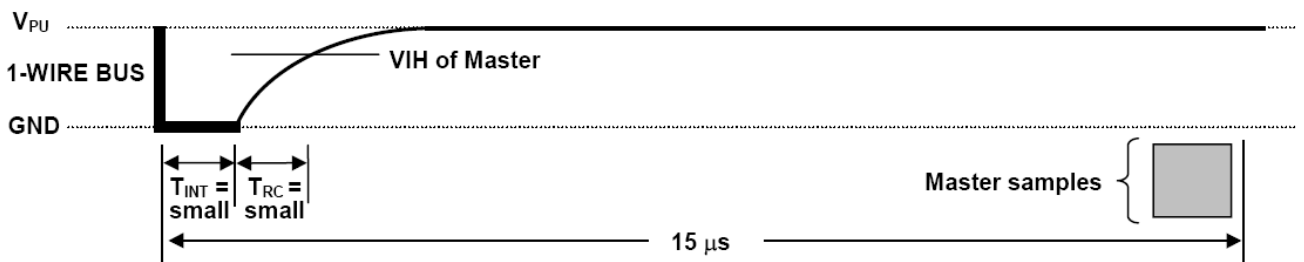
Все слоты времени считывания должны быть минимумом  $60\mu\text{s}$  и разделяться импульсами восстановления минимум  $1\mu\text{s}$  между слотами. Слот времени считывания инициализирован главным устройством, устанавливает на шине уровень логического нуля минимум на  $1\mu\text{s}$  и затем отпускает шину (см. иллюстрацию 14). После того, как хозяин инициализирует слот времени считывания, DS18B20 начнет передавать 1 или 0 на шине. DS18B20 передает 1, оставляя шину в высоком уровне и передает 0, устанавливая на шине 0. Выходные данные от DS18B20 достоверны через  $15\mu\text{s}$  после отрицательного уровня, который инициализировал слот времени считывания. Поэтому, хозяин должен выпустить шину и затем начать считывание шины не ранее  $15\mu\text{s}$  от начала слота.



Иллюстрация 15 иллюстрирует это сумма  $T_{\text{INIT}}$ ,  $T_{\text{RC}}$ , и  $T_{\text{SAMPLE}}$  должен быть меньше чем  $15\mu\text{s}$  для слота времени считывания. Показы иллюстрации 16, что система, рассчитывающая край развернута, сохраняя  $T_{\text{INIT}}$  и  $T_{\text{RC}}$ , почти как возможный и определяя местонахождение главного типового времени в течение слотов времени считывания к концу  $15\mu\text{s}$  период.

### DETAILED MASTER READ 1 TIMING Figure 14



### RECOMMENDED MASTER READ 1 TIMING Figure 15



LINE TYPE LEGEND	
	Bus master pulling low
	Resistor pullup

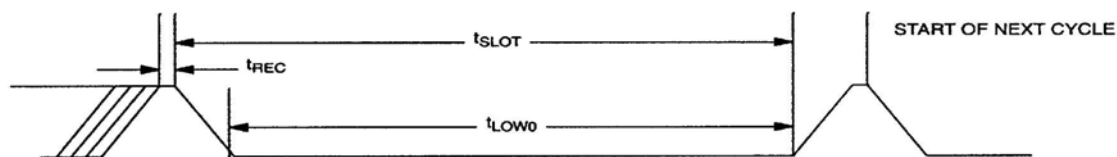
## AC ELECTRICAL CHARACTERISTICS (-55°C to +100°C; $V_{PU}=3.0V$ to 5.5V)

PARAMETER	SYMBOL	CONDITION	MIN	TYP	MAX	UNITS	NOTES
Temperature Conversion Time	$t_{CONV}$	9-bit resolution			93.75	ms	1
		10-bit resolution			187.5	ms	1
		11-bit resolution			375	ms	1
		12-bit resolution			750	ms	1
Time to Strong Pullup On	$t_{SPON}$	Start Convert T or Copy Scratchpad Command Issued			10	$\mu s$	
Time Slot	$t_{SLOT}$		60		120	$\mu s$	1
Recovery Time	$t_{REC}$		1			$\mu s$	1
Write 0 Low Time	$t_{LOW0}$		60		120	$\mu s$	1
Write 1 Low Time	$t_{LOW1}$		1		15	$\mu s$	1
Read Data Valid	$t_{RDV}$				15	$\mu s$	1
Reset Time High	$t_{RSTH}$		480			$\mu s$	1
Reset Time Low	$t_{RSTL}$		480		960	$\mu s$	1,2
Presence Detect High	$t_{PDHIGH}$		15		60	$\mu s$	1
Presence Detect Low	$t_{PDLOW}$		60		240	$\mu s$	1
Capacitance	$C_{IN/OUT}$				25	pF	

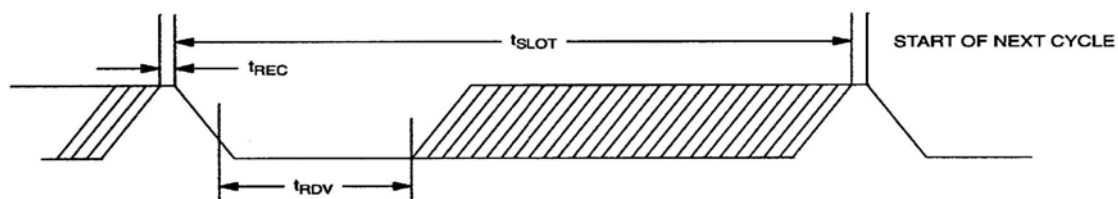
### NOTES:

1. Refer to timing diagrams in Figure 17.
2. If  $t_{RSTL} > 960 \mu s$ , a power on reset may occur.

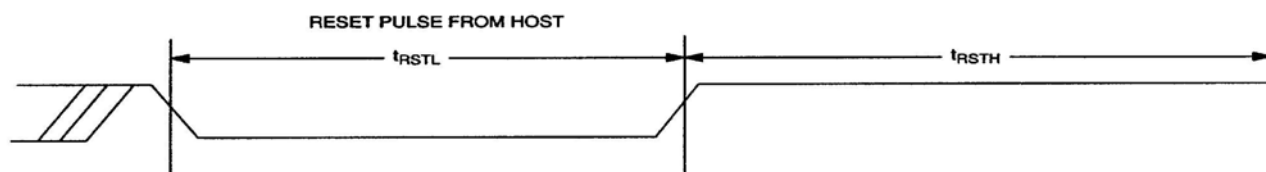
#### 1-WIRE WRITE ZERO TIME SLOT



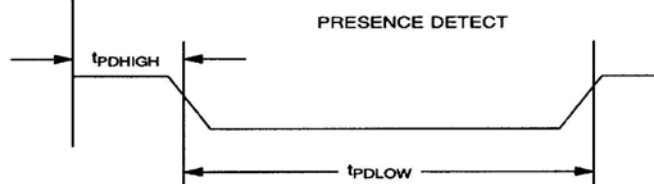
#### 1-WIRE READ ZERO TIME SLOT



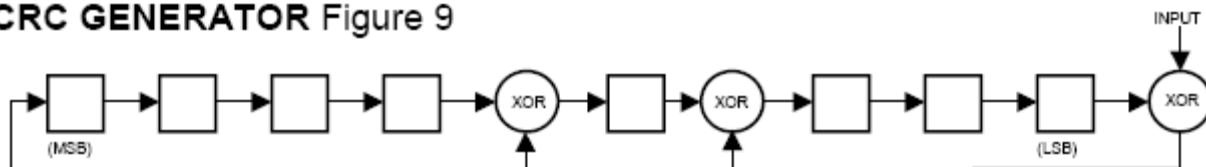
#### 1-WIRE RESET PULSE



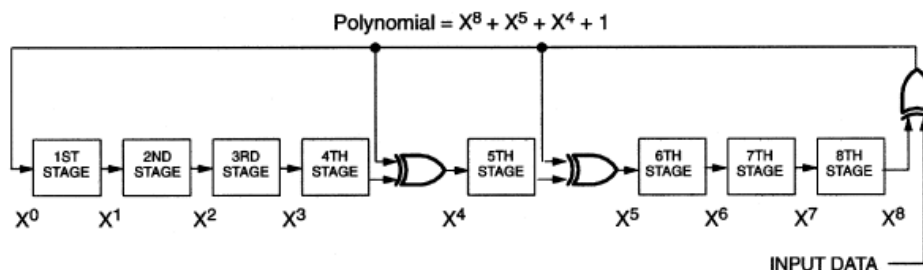
#### 1-WIRE PRESENCE DETECT



**CRC GENERATOR Figure 9**



**DALLAS 1-WIRE 8-BIT CRC Figure 2**



### Генерирование ЦИКЛИЧЕСКОГО КОНТРОЛЯ ИЗБЫТОЧНОСТИ

Байты циклического контроля избыточности обеспечиваются как часть 64-битового кода ROM DS18B20's и в 9-ом байте сверхоперативной памяти. Циклический контроль избыточности кода ROM вычислен от первых 56 битов кода ROM и содержится в наиболее значительном байте ROM. Сверхоперативный циклический контроль избыточности вычислен от данных, сохраненных в сверхоперативном, и поэтому это изменяется когда данные в сверхоперативных изменениях. Контроль с помощью циклического избыточного кода предоставляет устройству управления шиной метод проверки правильности данных, когда данные читаются от DS18B20. Чтобы проверять, что данные читались правильно, устройство управления шиной должно повторно вычислить циклический контроль избыточности от полученных данных и затем сравнить это значение с любым циклический контроль избыточности кода ROM (для чтений ROM) или к сверхоперативному циклическому контролю избыточности (для сверхоперативных чтений). Если расчетный циклический контроль избыточности соответствует циклическому контролю избыточности чтения, полученные данные были свободные от ошибок. Сравнение значений циклического контроля избыточности и решения продолжаться с операцией определено полностью устройством управления шиной. Нет никакой схемы в DS18B20, который препятствует последовательности команды продолжаться, если циклический контроль избыточности DS18B20 (ROM или сверхоперативный) не соответствует значению, сгенерированному устройством управления шиной.

Эквивалентная полиномиальная функция циклического контроля избыточности (ROM или сверхоперативный):

$$\text{CRC} = X^8 + X^5 + X^4 + 1$$

Устройство управления шиной может повторно вычислить циклический контроль избыточности и сравнить это со значениями циклического контроля избыточности от DS18B20, используя полиномиальный генератор, которому показывают в иллюстрации 9. Эта схема состоит из сдвигового регистра и Гейтса XOR, и биты сдвигового регистра инициализированы к 0. Старт с наименьшего значащего бита кода ROM или наименьшего значащего бита бита 0 в сверхоперативном, один бит одновременно должен сдвинутый в сдвиговый регистр. После смещения в 56-ом бите от ROM или наиболее значительного бита бита 7 от сверхоперативного, полиномиальный генератор будет содержать перерасчетный циклический контроль избыточности. Затем, 8-битовый код ROM или

сверхоперативный циклический контроль избыточности от DS18B20 должны быть сдвинуты в схему. В этом пункте, если перерасчетный циклический контроль избыточности был правилен, сдвиговый регистр будет содержать весь 0s. Дополнительная информация о Далласском циклическом контроле избыточности С 1 проводом 7 из 20 DS18B20 является доступным в Прикладном Примечании 27: Понимание и Используя Циклический контроль избыточности с Далласскими Продуктами Памяти Касания Полупроводника.

#### DS18B20 OPERATION EXAMPLE 1

В этом примере есть многократные DS18B20s на шине, и они используют паразитное питание. Устройство управления шиной инициализирует температурное преобразование в определенном DS18B20 и затем читает его сверхоперативное и повторно вычисляет циклический контроль избыточности, чтобы проверить данные.

MASTER MODE	DATA (LSB FIRST)	COMMENTS
TX	Reset	Хозяин выпускает импульс сброса.
RX	Presence	DS18B20s отвечают импульсом присутствия.
TX	55h	Хозяин выпускает команду ROM Соответствия.
TX	64-bit ROM code	Хозяин посылает код ROM DS18B20.
TX	44h	Хозяин выпускает команду Convert.
TX	DQ line held high by strong pullup	Хозяин применяет сильный pullup к способному к глубокой вытяжке для продолжительности преобразования (tconv).
TX	Reset	Хозяин выпускает импульс сброса.
RX	Presence	DS18B20s отвечают импульсом присутствия.
TX	55h	Хозяин выпускает команду ROM Соответствия.
TX	64-bit ROM code	Хозяин посылает код ROM DS18B20.
TX	BEh	Хозяин выпускает команду Read Scratchpad. (читать Память)
RX	9 data bytes	Главное устройство читает все регистры, включая сверхоперативный циклический контроль избыточности. Главное устройство вычисляет циклический контроль избыточности <b>первых восьми байтов данных</b> и сравнивает расчетный циклический контроль избыточности с циклическим контролем избыточности чтения (байт 9). Если они соответствуют, прием данных считается успешным, иначе ГУ повторяет цикл чтения.



## DS18B20 OPERATION EXAMPLE 2

В этом примере есть только один DS18B20 на шине, и это использует власть паразита. Хозяин пишет (TH, TL, and config). и регистрам конфигурации в сверхоперативном DS18B20 и затем читает сверхоперативное и повторно вычисляет циклический контроль избыточности, чтобы проверить данные. Хозяин тогда копирует сверхоперативное содержание на СППЗУ.

MASTER MODE	DATA (LSB FIRST)	COMMENTS
TX	Reset	Master issues reset pulse.
RX	Presence	DS18B20 responds with presence pulse.
TX	CCh	Master issues Skip ROM command.
TX	4Eh	Master issues Write Scratchpad command.
TX	3 data bytes	Master sends three data bytes to scratchpad (TH, TL, and config).
TX	Reset	Master issues reset pulse.
RX	Presence	DS18B20 responds with presence pulse.
TX	CCh	Master issues Skip ROM command.
TX	BEh	Master issues Read Scratchpad command.
RX	9 data bytes	Master reads entire scratchpad including CRC. The master then recalculates the CRC of the first eight data bytes from the scratchpad and compares the calculated CRC with the read CRC (byte 9). If they match, the master continues; if not, the read operation is repeated.
TX	Reset	Master issues reset pulse.
RX	Presence	DS18B20 responds with presence pulse.
TX	CCh	Master issues Skip ROM command.
TX	48h	Master issues Copy Scratchpad command.
TX	DQ line held high by strong pullup	Master applies strong pullup to DQ for at least 10ms while copy operation is in progress.