

AD9833: программируемый генератор сигналов



Добавил(а) microsin

Основные возможности микросхемы:

- Программируемая через цифровой интерфейс частота и фаза.
- Низкое энергопотребление: 12.65 мВт при напряжении питания 3V.
- Диапазон выходной частоты от 0 до 12.5 МГц.
- 28-битная разрешающая способность по частоте: 0.1 Гц при опорной частоте тактов 25 МГц.
- Форма выходного сигнала может быть синусоидальная, треугольная и прямоугольная.
- Диапазон напряжений питания от 2.3 до 5.5V.
- Не требуется внешних компонентов (почти).
- 3-проводный интерфейс SPI.
- Расширенный температурный диапазон -40°C .. +105°C.
- Возможность выключения (Power-down).
- 10-выводный корпус типа MSOP.
- Дизайн рассчитан на мобильные, автономные приложения.

Области применения:

- Генерация сигналов и тактовых частот, генераторы качающейся частоты.
- Системы измерения потока жидкости и газа.
- Приложения сенсоров: сближение, движение, дефектоскопия.
- Потеря линии / подстройка под параметры линии.
- Оборудование тестирования, медицина.
- Приложения временной доменной рефлектометрии (Time domain reflectometry, TDR).

[Общее описание]

Микросхема AD9833 это программируемый генератор сигналов с низким потреблением тока, который может генерировать сигнал синусоидальной, треугольной и прямоугольной формы. Генерация сигнала определенной формы требуется для разных типов реализаций датчиков, актуаторов, приложений TDR. Выходная частота и фаза программируется под управлением внешнего микроконтроллера, что упрощает настройку генерации. Для запуска микросхемы в работу почти не требуется внешних компонентов (кроме внешнего генератора опорных тактов и блокирующих фильтрующих конденсаторов). Регистры частоты имеют разрядность 28 бит: при тактовой частоте 25 МГц может быть достигнута дискретность установки частоты 0.1 Гц; с тактовой частотой 1 МГц может быть достигнута дискретность 0.004 Гц.

Параметры записываются в AD9833 через 3-проводный последовательный интерфейс (SPI). Этот последовательный интерфейс может работать на частотах до 40 МГц, и он хорошо подходит для подключения к любому DSP и микроконтроллеру. Микросхема может работать от напряжения источника питания 2.3 .. 5.5V.

У AD9833 есть функция выключения (power-down), или режим сна, функция SLEEP. Эта функция позволяет отключить не используемые секции устройства, минимизируя тем самым потребление тока. Например, DAC (ЦАП) может быть выключен, когда генерируется тактовый выходной сигнал (прямоугольная форма сигнала на выходе).

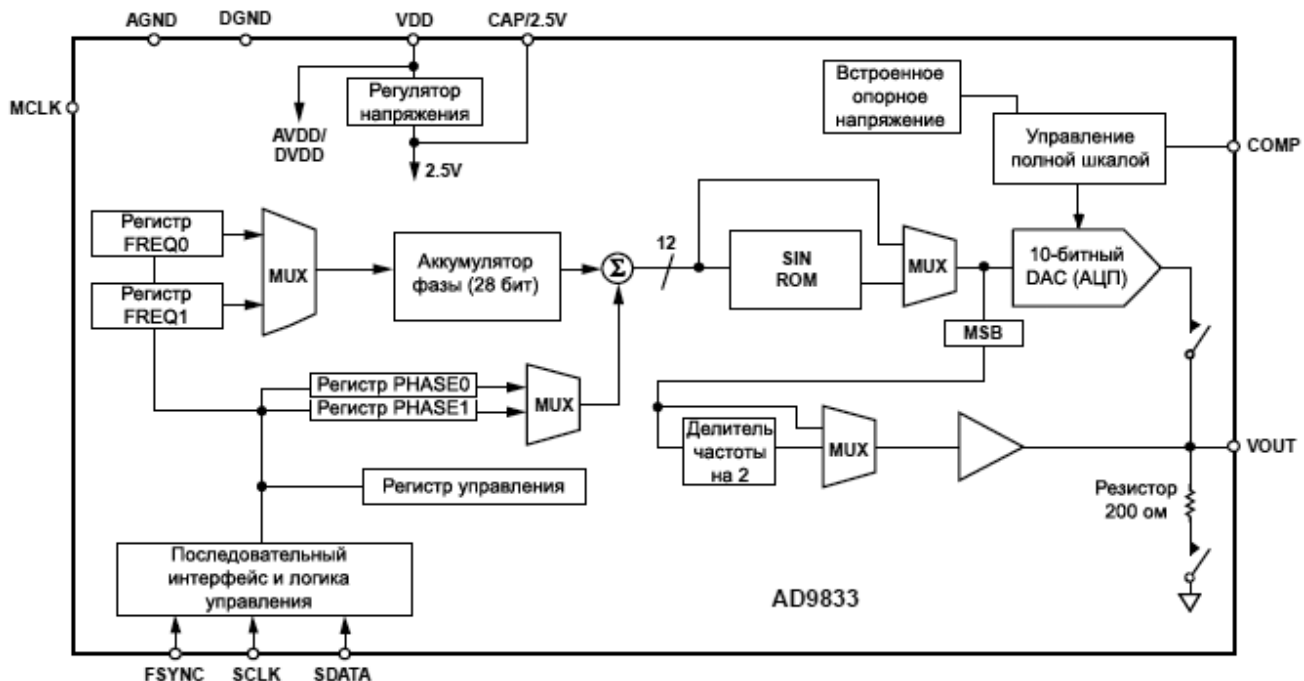


Рис. 1. Функциональная блок-диаграмма AD9833.

[Подобные микросхемы, оценочные киты и документация](#)

Микросхемы DDS, предназначенные для цифрового синтеза сигналов (подобные по функционалу AD9833).

Оценочный кит разработчика (замечание: у китайцев на Aliexpress можно найти платы получше).

Документация, апноуты, даташиты, книги, руководства:

- AN-1044: Programming the AD5932 for Frequency Sweep and Single Frequency Outputs
- AN-1070: Programming the AD9833/AD9834
- AN-1248: SPI Interface
- AN-1389: Recommended Rework Procedure for the Lead Frame Chip Scale Package (LFCSP)
- AN-237: Choosing DACs for Direct Digital Synthesis
- AN-280: Mixed Signal Circuit Technologies
- AN-342: Analog Signal-Handling for High Speed and Accuracy
- AN-345: Grounding for Low-and-High-Frequency Circuits
- AN-419: A Discrete, Low Phase Noise, 125 MHz Crystal Oscillator for the AD9850
- AN-423: Amplitude Modulation of the AD9850 Direct Digital Synthesizer
- AN-543: High Quality, All-Digital RF Frequency Modulation Generation with the ADSP-2181 and the AD9850 DDS
- AN-557: An Experimenter's Project
- AN-587: Synchronizing Multiple AD9850/AD9851 DDSBased Synthesizers
- AN-605: Synchronizing Multiple AD9852 DDS-Based Synthesizers
- AN-621: Programming the AD9832/AD9835
- AN-632: Provisionary Data Rates Using the AD9951 DDS as an Agile Reference Clock for the ADN2812 Continuous-Rate CDR
- AN-769: Generating Multiple Clock Outputs from the AD9540
- AN-772: A Design and Manufacturing Guide for the Lead Frame Chip Scale Package (LFCSP)
- AN-823: Direct Digital Synthesizers in Clocking Applications Time • AN-837: DDS-Based Clock Jitter Performance vs. DAC Reconstruction Filter Performance
- AN-843: Measuring a Loudspeaker Impedance Profile Using the AD5933
- AN-847: Measuring a Grounded Impedance Profile Using the AD5933
- AN-851: A WiMax Double Downconversion IF Sampling Receiver Design
- AN-927: Determining if a Spur is Related to the DDS/DAC or to Some Other Source (For Example, Switching Supplies)
- AN-939: Super-Nyquist Operation of the AD9912 Yields a High RF Output Signal
- AN-953: Direct Digital Synthesis (DDS) with a Programmable Modulus
- Introducing Digital Up/Down Converters: VersaCOMM™ Reconfigurable Digital Converters
- A Technical Tutorial on Digital Signal Synthesis, 1999
- UG-272: Evaluating the AD9833 Low Power 12.65 mW, 2.3 V to 5.5 V, Programmable Waveform Generator

Программное обеспечение и системные требования

- AD9833 - Microcontroller No-OS Driver
- AD9834 IIO Direct Digital Synthesis Linux Driver
- AD9833 Evaluation Board Software
- AD9833 FMC-SDP Interposer & Evaluation Board / Xilinx KC705 Reference Design
- BeMicro FPGA Project for AD9833 with Nios driver

Инструментарий и симуляция:

- ADIsimDDS (Direct Digital Synthesis)
- AD9833 IBIS Model

Материалы по теме, технические статьи:

- 400-MSample DDSs Run On Only +1.8 VDC
- ADI Buys Korean Mobile TV Chip Maker
- Basics of Designing a Digital Radio Receiver (Radio 101)
- DDS Applications
- DDS Circuit Generates Precise PWM Waveforms
- DDS Design
- DDS Device Produces Sawtooth Waveform
- DDS Device Provides Amplitude Modulation
- DDS IC Initiates Synchronized Signals
- DDS IC Plus Frequency-To-Voltage Converter Make Low-Cost DAC
- DDS Simplifies Polar Modulation
- Digital Potentiometers Vary Amplitude In DDS Devices
- Digital Up/Down Converters: VersaCOMM™ White Paper
- Digital Waveform Generator Provides Flexible Frequency Tuning for Sensor Measurement
- Improved DDS Devices Enable Advanced Comm Systems
- Integrated DDS Chip Takes Steps To 2.7 GHz
- Simple Circuit Controls Stepper Motors
- Speedy A/Ds Demand Stable Clocks
- Synchronized Synthesizers Aid Multichannel Systems
- The Year of the Waveform Generator
- Two DDS ICs Implement Amplitude-shift Keying
- Video Portables and Cameras Get HDMI Outputs

Материалы по дизайну:

- AD9833 Material Declaration
- PCN-PDN Information
- Quality And Reliability
- Symbols and Footprints

Обсуждения AD9833 на EngineerZone

Таблица 1. Электрические и сигнальные параметры.

Если не указано нечто обратное, то параметры в таблице даны для условий $V_{DD} = 2.3 \dots 5.5V$, $AGND = DGND = 0V$, $T_A = T_{MIN} \dots T_{MAX}$, $R_{SET} = 6.8 \text{ кОм}$ для V_{OUT} .

Параметр ¹	MIN	Типично	MAX	Ед.	Условия тестирования / комментарии
<i>Спецификации сигнала DAC</i>					
Разрешающая способность		10		бит	
Скорость обновления			25	MSPS ²	
Максимальное значение V_{OUT}		0.65		V	
Минимальное значение V_{OUT}		38		mV	
Температурный коэффициент V_{OUT}		200		ppm/ °C ³	
Точность по постоянному напряжению (DC), интегральная нелинейность (INL) ⁵		±1.0		LSB ⁴	
Точность по постоянному напряжению (DC), дифференциальная нелинейность (DNL) ⁵		±0.5		LSB ⁴	
<i>Спецификации DDS</i>					
Динамика, SNR ⁵	55	60		dB	$f_{MCLK} = 25 \text{ МГц}$, $f_{OUT} = f_{MCLK}/4096$
Динамика, THD ⁵		-66	-56	dBc ⁶	$f_{MCLK} = 25 \text{ МГц}$, $f_{OUT} = f_{MCLK}/4096$
SFDR ⁵ в широком диапазоне (0 к частоте Найквиста)		-60		dBc	$f_{MCLK} = 25 \text{ МГц}$, $f_{OUT} = f_{MCLK}/50$
SFDR в узком диапазоне ($\pm 200 \text{ кГц}$)		-78		dBc	$f_{MCLK} = 25 \text{ МГц}$, $f_{OUT} = f_{MCLK}/50$
Сквозное прохождение тактов		-60		dBc	
Время пробуждения (выход из Power Down)		1		мс	
<i>Логические входы</i>					
Входное напряжение лог. 1, V_{INH}	1.7			V	Питание 2.3 .. 2.7V
	2.0			V	Питание 2.7 .. 3.6V
	2.8			V	Питание 4.5 .. 5.5V
Входное напряжение лог. 0, V_{INL}			0.5	V	Питание 2.3 .. 2.7V
			0.7	V	Питание 2.7 .. 3.6V
			0.8	V	Питание 4.5 .. 5.5V
Входной ток, I_{INH}/I_{INL}			10	µA	
Входная емкость, C_{IN}		3			пФ
<i>Питание ($f_{MCLK} = 25 \text{ МГц}$, $f_{OUT} = f_{MCLK}/4096$)</i>					
V_{DD}	2.3		5.5	V	
I_{DD}		4.5		mA	I_{DD} зависит от кода, см. рис. 7
Режим сна с пониженным энергопотреблением (Low Power Sleep Mode)		0.5		mA	DAC выключен, тактирование MCLK активно

Примечания:

1 Рабочий диапазон температур $-40^{\circ}\text{C} \dots +105^{\circ}\text{C}$; типичные спецификации подразумевают $+25^{\circ}\text{C}$.

2 MSPS означает Million Samples Per Second, т. е. миллион выборок в секунду.

3 ppm означает parts per million.

4. LSB означает Least Significant Bit, т. е. имеется в виду вес самого младшего разряда DAC.

5. Описание терминологии см. в соответствующей врезке ниже.

6. dBc это соотношение мощности сигнала к сигналу несущей, выраженное в децибелах (dB). Например, фазовый шум выражается в dBc/Гц на указанном смещении частоты от несущей. dBc может также использоваться в измерении свободного от помех динамического диапазона (Spurious-Free Dynamic Range, SFDR) между желаемым сигналом и не желательными помехами, которые производят преобразователи DAC или частотные смесители (из Википедии).

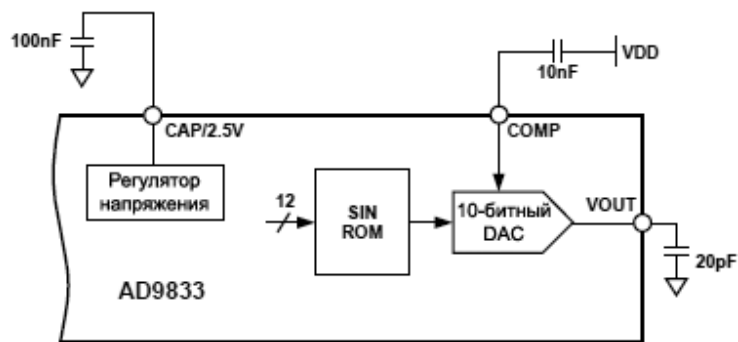


Рис. 2. Схема тестирования, примененная для тестовых спецификаций.

Таблица 2. Параметры времени.

Если не указано нечто обратное¹, то параметры в таблице даны для условий $VDD = 2.3 \dots 5.5V$, $AGND = DGND = 0V$.

Параметр	Предел между TMIN и TMAX	Ед.	Описание
t_1	40	нс min	Период MCLK
t_2	16	нс min	Длительность лог. 1 MCLK
t_3	16	нс min	Длительность лог. 0 MCLK
t_4	25	нс min	Период SCLK
t_5	10	нс min	Длительность лог. 1 SCLK
t_6	10	нс min	Длительность лог. 0 SCLK
t_7	5	нс min	Время установки, интервал между спадом FSYNC и первым спадом SCLK
t_{8min}	10	нс min	
t_{8max}	$t_4 - 5$	нс max	Время удержания FSYNC для SCLK
t_9	5	нс min	Время установки данных
t_{10}	3	нс min	Время удержания данных
t_{11}	5	нс min	Время установки, интервал между SCLK в лог. 1 и спадом FSYNC

Примечание 1: гарантировано дизайном, не тестируется в производстве.

Диаграммы времени

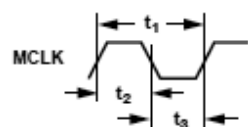


Рис. 3. Главная тактовая частота (Master Clock).

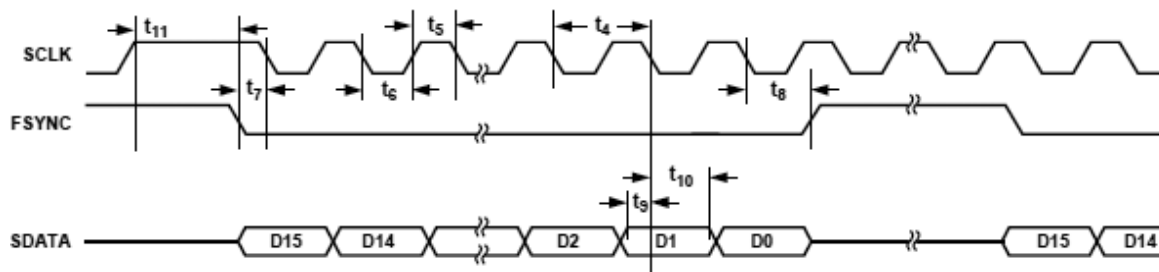


Рис. 4. Диаграмма сигналов последовательного интерфейса.

Таблица 3. Предельно допустимые параметры.

$T_A = 25^\circ C$, если не указано нечто иное.

Параметр	Значение
VDD относительно AGND	$-0.3V \dots +6V$
VDD относительно DGND	$-0.3V \dots +6V$
AGND относительно DGND	$-0.3V \dots +0.3V$

CAP/2.5V	2.75 V
Напряжение цифрового ввода/вывода относительно DGND	−0.3 V .. (VDD + 0.3 V)
Напряжение аналогового ввода/вывода относительно AGND	−0.3 V .. (VDD + 0.3 V)
Рабочий диапазон температур промышленного применения (версия B)	−40°C .. +105°C
Диапазон температур хранения	−65°C .. +150°C
Максимальная температура кристалла (Maximum Junction Temperature)	150°C
Корпус MSOP	

[Цоколевка и функции выводов]



Рис. 5. Цоколевка выводов корпуса MSOP10.

Таблица 4. Функциональное описание выводов.

№	Мнемоника	Описание
1	COMP	Вывод смещения АЦП (DAC Bias). Этот вывод используется для подключения блокировочного конденсатора, развязывающего от помех напряжение смещения АЦП.
2	VDD	Положительное напряжение питания для аналоговых секций схемы и цифрового интерфейса. Встроенный регулятор 2.5 V также получает питание от VDD. VDD может иметь значение от 2.3 V до 5.5 V. Развязывающие конденсаторы 0.1 мкФ и 10 мкФ должны быть подключены между VDD и AGND.
3	CAP/2.5V	Цифровая схемотехника работает от напряжения питания 2.5 V. Это напряжение генерируется из VDD с помощью встроенного регулятора, когда VDD превышает 2.7 V. Регулятор требует подключения к своему выходу развязывающего конденсатора типового номинала 100 нФ, который подключается между выводом CAP/2.5V и DGND. Если VDD меньше или равно 2.7 V, вывод CAP/2.5V должен быть напрямую подключен к VDD.
4	DGND	Цифровая земля.
5	MCLK	Вход основной частоты тактов (Master Clock, MCLK). Выходные частоты DDS выражаются как двоичное делимое от частоты MCLK. Этой частой определяется точность выходной частоты и фазовый шум на выходе.
6	SDATA	Последовательный вход данных (SPI). На этот вход приходит 16-битное слово данных.
7	SCLK	Вход последовательных тактов. Данные SDATA для AD9833 защелкиваются по каждому спаду сигнала SCLK.
8	FSYNC	Вход управления цифровым интерфейсом (выборка с активным лог. 0). FSYNC является синхронизирующим сигналом фрейма для входных данных. Когда FSYNC == 0, внутренняя логика интерфейса информируется о том, что в микросхему загружается новое слово.
9	AGND	Аналоговая земля.
10	VOUT	Напряжение выходного сигнала. Здесь формируется аналоговый и цифровой выход (в зависимости от режима) сигнала. Внешняя нагрузка на этом выводе не требуется, потому что в микросхеме имеется встроенный резистор нагрузки 200 Ω.

Графики типовых характеристик

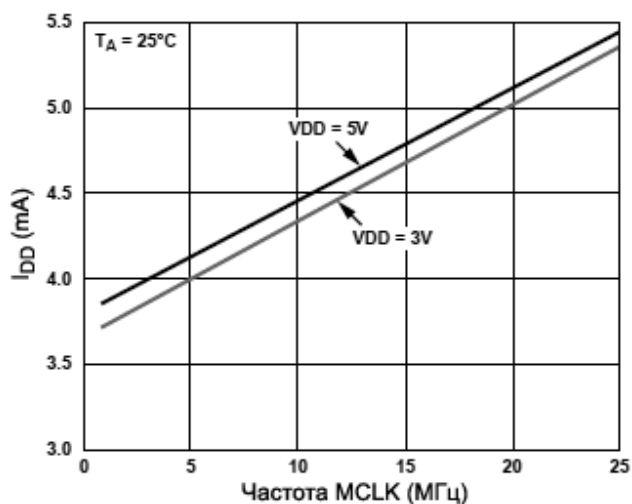


Рис. 6. Типичное потребление тока (I_{DD}) в зависимости от частоты MCLK для $f_{OUT} = MCLK/10$

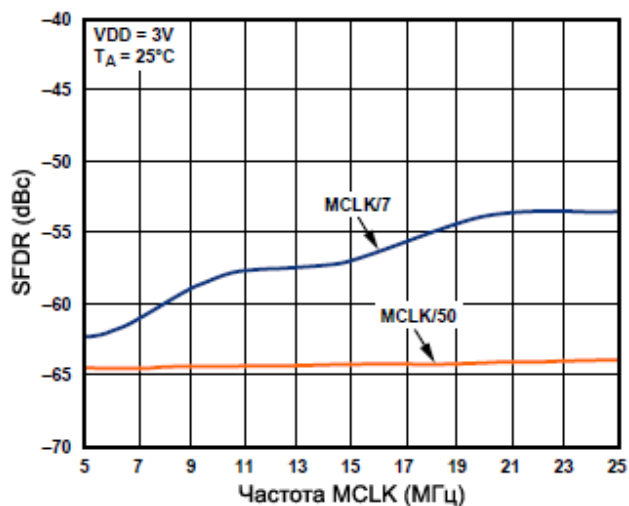


Рис. 9. Широкополосный SFDR в зависимости от частоты MCLK

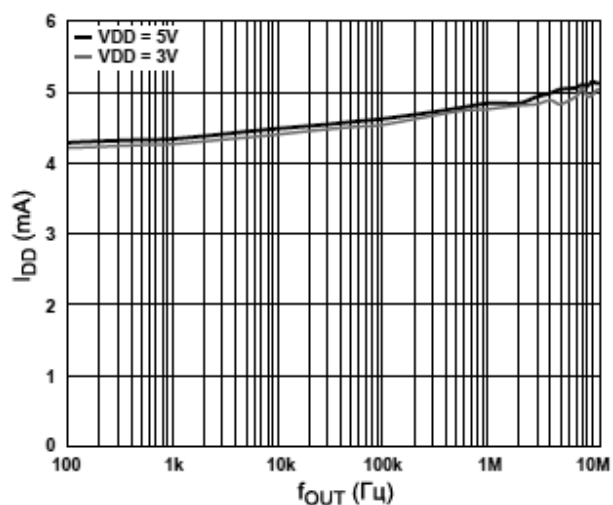


Рис. 7. Типичный I_{DD} в зависимости от f_{OUT} для $f_{MCLK} = 25$ МГц

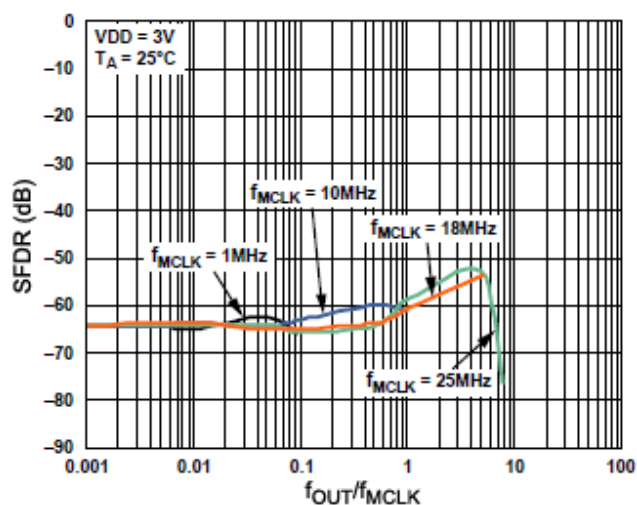


Рис. 10. Широкополосный SFDR в зависимости от f_{OUT}/f_{MCLK} для различных частот MCLK

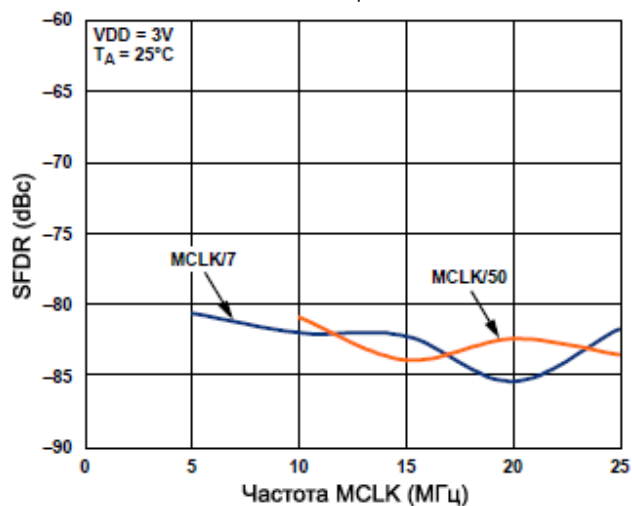


Рис. 8. Узкополосный SFDR в зависимости от частоты MCLK

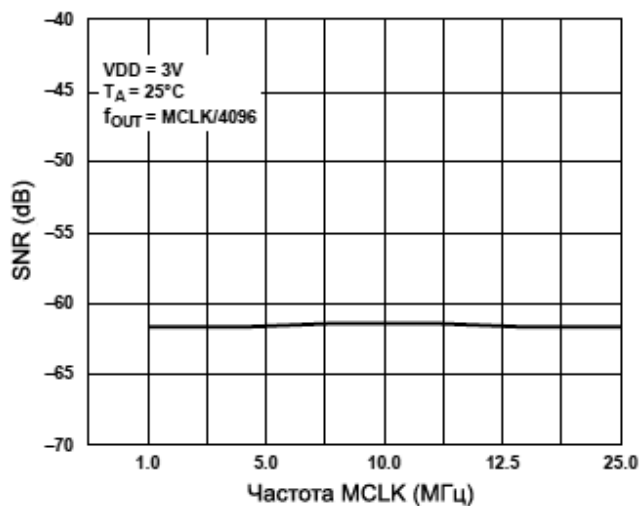


Рис. 11. SNR в зависимости от частоты MCLK

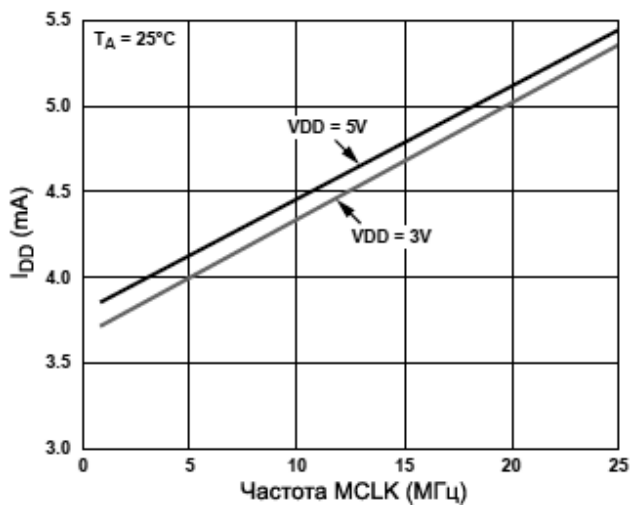


Рис. 12. Время пробуждения (Wake-Up Time) в зависимости от температуры

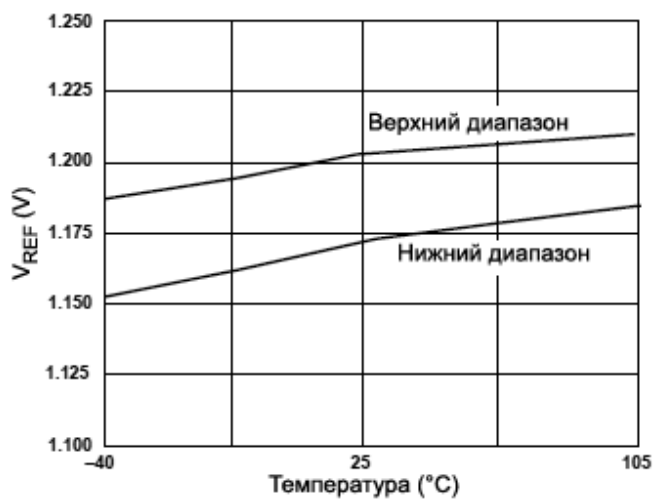


Рис. 13. V_{REF} в зависимости от температуры

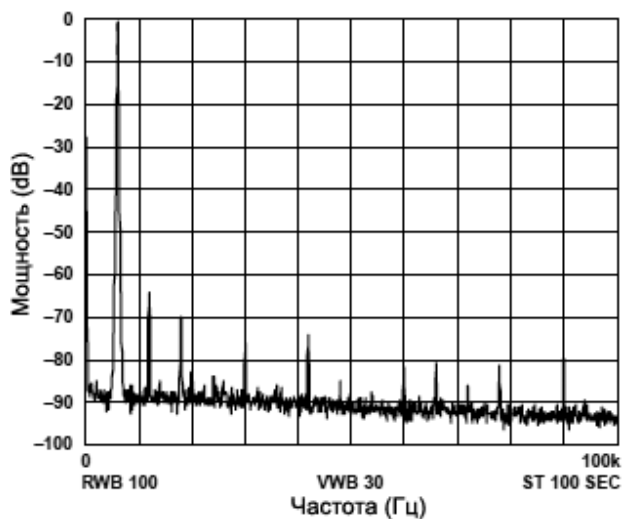


Рис. 14. Зависимость выходной мощности от частоты, $f_{MCLK} = 10$ МГц, $f_{OUT} = 2.4$ кГц = $f_{MCLK}/3$, слово частоты = 0x000FBA9

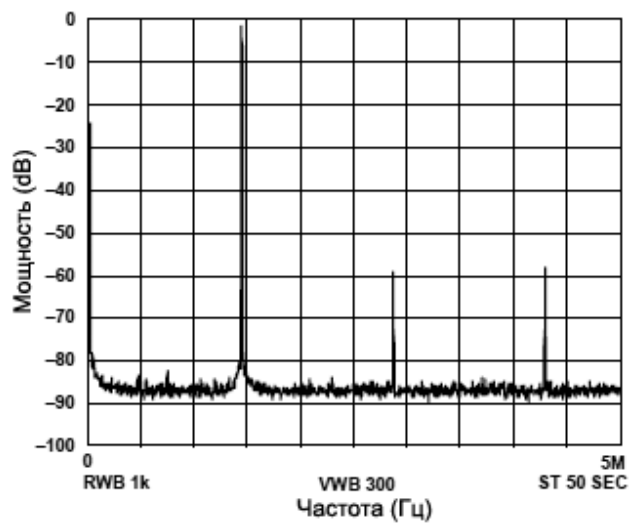


Рис. 15. Зависимость выходной мощности от частоты, $f_{MCLK} = 10$ МГц, $f_{OUT} = 1.43$ МГц = $f_{MCLK}/7$, слово частоты = 0x2492492

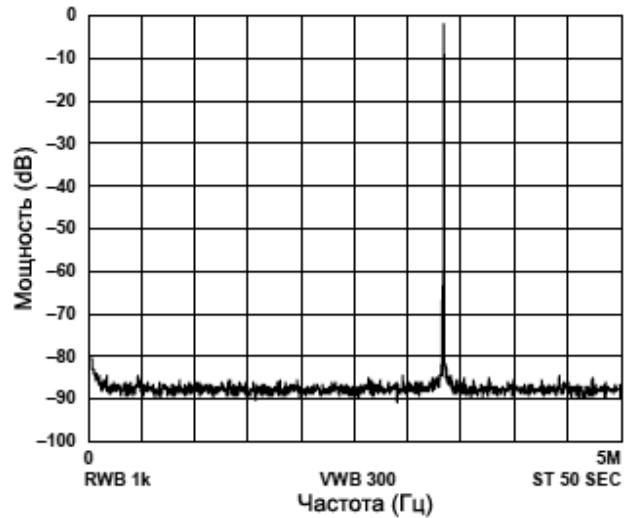


Рис. 16. Зависимость выходной мощности от частоты, $f_{MCLK} = 10$ МГц, $f_{OUT} = 3.33$ МГц = $f_{MCLK}/3$, слово частоты = 0x5555555

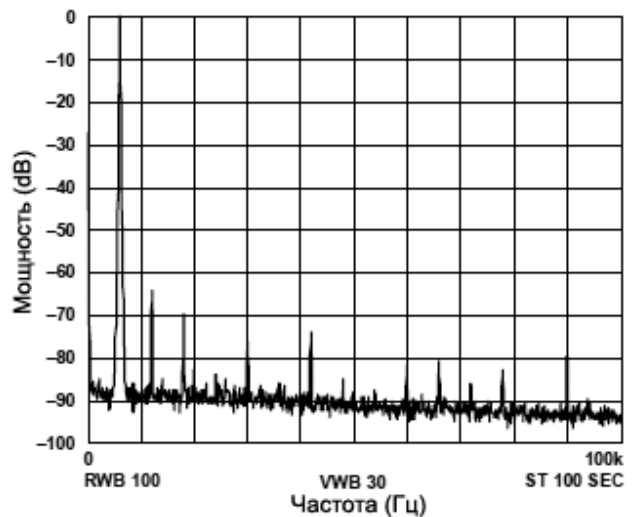


Рис. 17. Зависимость выходной мощности от частоты, $f_{MCLK} = 25$ МГц, $f_{OUT} = 6$ кГц, слово частоты = 0x000FBA9

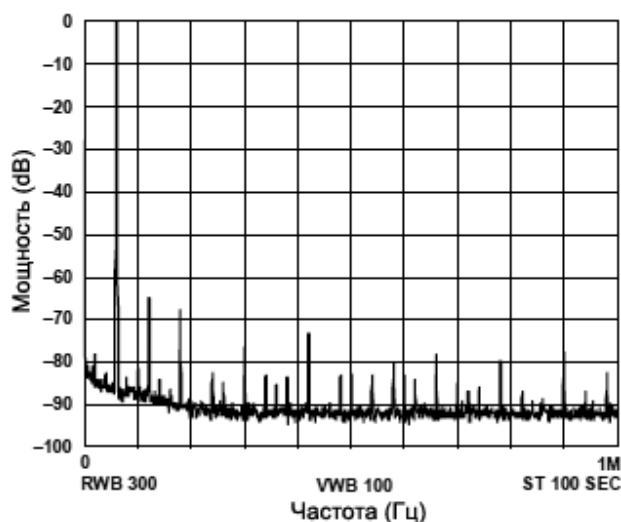


Рис. 18. Зависимость выходной мощности от частоты, $f_{\text{MCLK}} = 25 \text{ МГц}$, $f_{\text{OUT}} = 60 \text{ кГц}$, слово частоты = 0x009D495

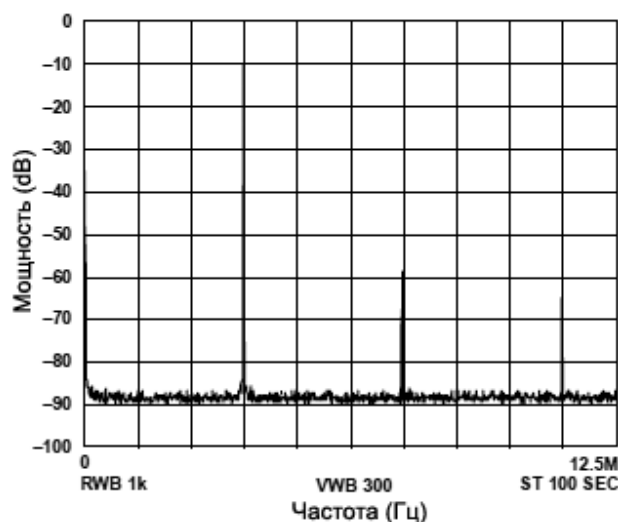
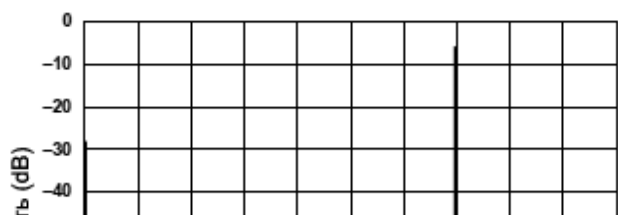
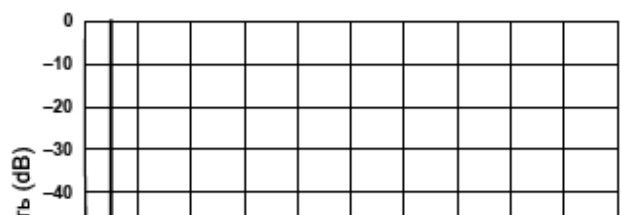


Рис. 21. Зависимость выходной мощности от частоты, $f_{\text{MCLK}} = 25 \text{ МГц}$, $f_{\text{OUT}} = 3.857 \text{ МГц} = f_{\text{MCLK}}/7$, слово частоты = 0x2492492



Используемая терминология

Интегральная нелинейность. Integral Nonlinearity (INL) это максимальная девиация любого кода от прямой линии, проходящая через конечные точки передаточной функции. Конечные точки передаточной функции - нулевой масштаб, точка 0.5 LSB ниже первого изменения кода (от 000 ... 00 к 000 ... 01), и полный масштаб, точка 0.5 LSB выше последнего перехода кода (от 111 ... 10 к 111 ... 11). Ошибка выражается в количестве разрядов LSB.

Дифференциальная нелинейность. Differential Nonlinearity (DNL) это разница между измеренным и идеальным изменением 1 LSB между двумя соседними кодами DAC. Указанная DNL $\pm 1 \text{ LSB}$ максимум гарантирует монотонность характеристики.

Совместимость по выходу (Output Compliance). Этот параметр относится к выходному напряжению, которое может быть сгенерировано до максимального на выходе, когда DAC удовлетворяет спецификациям. Когда напряжение больше, чем указано для совместимости по выходу, AD9833 может не удовлетворять спецификациям, перечисленным в даташите.

Динамический диапазон без помех. Spurious-Free Dynamic Range (SFDR) - вместе с интересующей частотой на выходе устройства DDS присутствуют также гармоники основной частоты и образы этих частот. SFDR относится к самой большой побочной составляющей сигнала или гармонике, находящейся в интересующем диапазоне сигнала. Широкополосный SFDR дает амплитуду самой большой помехи или гармоники относительно амплитуды фундаментальной частоты у нуля к полосе Найквиста. Узкополосный SFDR дает подстройку самой большой гармоники или помехи в полосе частот $\pm 200 \text{ кГц}$ относительно основной частоты.

Среднее квадратическое значение сигнала. Root Mean Square (RMS) - характеристика сигнала переменного тока, которая определяет его мощность. Среднеквадратичное значение переменного тока равно величине

постоянного тока, действие которого произведёт такую же работу в активной (резистивной) нагрузке за время периода.

Коэффициент общих гармонических искажений. Total Harmonic Distortion (THD) это соотношение rms суммы всех гармоник на выходе к rms основной частоты. Для AD9833 THD определяется по формуле:

$$THD = 20 \log \sqrt{\frac{V_2^2 + V_3^2 + V_4^2 + V_5^2 + V_6^2}{V_1^2}}$$

Здесь:

V_1 это rms амплитуды основной частоты.

V_2, V_3, V_4, V_5 и V_6 rms амплитуд гармоник от второй до шестой.

Соотношение сигнал/шум. Signal-to-Noise Ratio (SNR) это отношение значения rms сигнала, измеренное на выходе, к сумме всех других побочных спектральных компонентов ниже частоты Найквиста. Значение SNR выражается в децибелах.

Сквозное прохождение частоты тактирования. Clock Feedthrough - характеризует прохождение помехи со входа MCLK на аналоговый выход. Clock feedthrough относится амплитуде сигнала MCLK относительно основной частоты в выходном спектре сигнала AD9833.

[Теория функционирования]

Синусоидальные сигналы обычно представляют с точки зрения их амплитуды: $a(t) = \sin(\omega t)$. Однако эти синусоидальные сигналы нелинейны, и их непросто генерировать, кроме как строить по кусочкам. С другой стороны, угловая информация по своей природе линейна. Таким образом, для определенной частоты угол фазы поворачивается на фиксированный угол для каждой единицы времени. Скорость изменения угла зависит от частоты сигнала по традиционному соотношению $\omega = 2\pi f$.

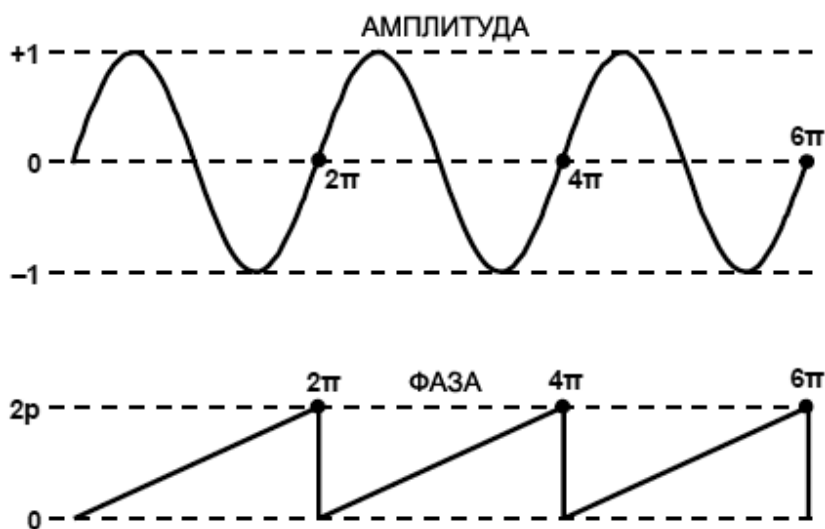


Рис. 23. Синусоидальный сигнал и закон изменения его фазы.

Зная о том, что фаза синусоидального сигнала линейна на указанном интервале времени (периоде тактирования), можно определить поворот фазы для этого периода.

$$\Delta \text{Phase} = \omega \Delta t$$

Угол ω вычисляется как

$$\omega = \Delta\text{Phase}/\Delta t = 2\pi f$$

Если вычислить f и подставить опорную частоту для опорного периода ($1/f_{\text{MCLK}} = \Delta t$), то получится формула

$$f = \Delta\text{Phase} * f_{\text{MCLK}} / 2\pi$$

AD9833 строит свой выходной сигнал на основе этого простого выражения. Простой чип DDS может реализовать это выражение тремя основными промежуточными узлами: управляемый через цифровой интерфейс генератор (numerically controlled oscillator, NCO) и модулятор фазы, ПЗУ синуса (SIN ROM) и цифро-аналоговый преобразователь (digital-to-analog converter, DAC).

Каждый из этих узлов описан в разделе "Описание внутренней схемы AD9833".

[Описание внутренней схемы AD9833]

AD9833 это полностью интегрированный чип прямого цифрового синтеза (direct digital synthesis, DDS). Для его работы требуется одна опорная частота, один резистор не высокой точности, и развязывающие конденсаторы, чтобы его схема синтезировала синусоидальный сигнал с частотой до 12.5 МГц. Дополнительно чип может генерировать широкий диапазон простых и сложных схем модуляции. Эти схемы модуляции полностью реализованы в цифровом домене, позволяя точно и просто реализовать сложные алгоритмы модуляции, используя техники DSP.

Внутренняя схема AD9833 состоит из следующих основных частей: управляемый числами генератор (NCO), модуляторы частоты и фазы, SIN ROM, DAC и регулятор.

NCO плюс модулятор фазы. Эта часть схемы состоит из двух регистров выбора частоты, аккумулятора фазы, двух регистров смещения фазы и сумматора смещения фазы. Основной компоненты NCO это 28-битный аккумулятор фазы. Непрерывные по времени сигналы работают с диапазоном изменения фазы от 0 до 2π . Вне этого диапазона чисел синусоидальные функции повторяют сами себя периодически. Цифровая реализация ничем не отличается. Аккумулятор просто масштабирует диапазон значений фазы в многобитное цифровое слово. Аккумулятор фазы в AD9833 реализован как 28-разрядный счетчик. Таким образом в AD9833 период $2\pi = 2^{28}$. Подобным образом понятие ΔPhase масштабируется в этот диапазон чисел:

$$0 < \Delta\text{Phase} < 2^{28} - 1$$

С этими подстановками предыдущее выражение получается в таком виде:

$$f = \Delta\text{Phase} * f_{\text{MCLK}} / 2^{28}$$

$$\text{где } 0 < \Delta\text{Phase} < 2^{28} - 1.$$

Вход аккумулятора фазы может быть выбран или из регистра FREQ0, или из регистра FREQ1, что управляется битом FSELECT. Генераторы NCO по сути формируют непрерывные по фазе сигналы, поэтому амплитуда на выходе остается неразрывной при переключениях между частотами.

Следуя за NCO, можно добавить смещение фазы, чтобы реализовать модуляцию фазы, используя 12-битные регистры фазы. Содержимое одного из этих регистров добавляется к самым старшим битам NCO. У AD9833 есть 2 регистра фазы; их разрешающая способность составляет $2\pi/4096$.

SIN ROM. Чтобы сделать выход NCO полезным для использования, он должен быть преобразован в синусоидальное значение. Поскольку информация фазы прямо привязана к амплитуде, SIN ROM использует эту цифровую информацию фазы в качестве адреса в таблице, и таким способом преобразует информацию фазы в амплитуду. Хотя NCO имеет 28-битный аккумулятор фазы, выход NCO обрезается до 12 бит. С использованием полного разрешения аккумулятора фазы непрактично, и это не требуется, поскольку это потребовало бы таблицы из 2^{28} записей. Требуется только такая разрешающая способность по фазе, чтобы исключить ошибки усечения из-за разрешающей способности меньше, чем разрешающая способность 10-битного DAC. Это требует, чтобы SIN ROM имело на 2 бита больше разрешающей способности, чем 10-битный DAC (т. е. 12 бит).

SIN ROM разрешается с использованием бита режима (D1) в регистре управления (см. таблицу 15).

DAC. Внутри AD9833 имеется 10-битный DAC (цифро-аналоговый преобразователь, ЦАП), работающий на основе источника тока с высоким сопротивлением. DAC получает цифровые данные из SIN ROM, и преобразует их в соответствующие аналоговые напряжения.

DAC сконфигурирован для работы в асимметричном режиме. Внешний нагрузочный резистор не требуется, потому что устройство имеет встроенный резистор 200 Ω . DAC генерирует выходное напряжение типичного уровня 0.6V от пика до пика.

Регулятор. VDD предоставляет питание, требуемое для аналоговой части и цифровой части AD9833. Это питание может иметь значение напряжения от 2.3 V до 5.5 V.

Внутренняя цифровая секция AD9833 работает от напряжения 2.5V. Внутренний встроенный регулятор понижает напряжение, прикладываемое к VDD, до уровня 2.5V. Когда внешнее прикладываемое к VDD напряжение меньше или равно 2.7V, выводы CAP/2.5V и VDD должны быть соединены друг с другом, это пропускает использование встроенного регулятора.

[Функциональное описание]

Последовательный интерфейс. AD9833 подключается через стандартный 3-проводный последовательный интерфейс, совместимый со стандартами SPI, QSPI™, MICROWIRE® и интерфейсами цифровых процессоров DSP.

Данные загружаются в устройство как 16-битное слово под управлением последовательных тактов вывода SCLK. Форма сигналов интерфейса и диаграммы времени показаны на рис. 4.

Вход FSYNC работает по уровню как синхронизация фрейма и разрешающий сигнал интерфейса (chip enable). Данные могут быть переданы в устройство только если FSYNC находится в состоянии лог. 0. Для запуска последовательной передачи FSYNC нужно перевести в лог. 0, соблюдая минимальное время установки после спада FSYNC, интервал между спадом FSYNC до спада SCLK, время t_7 . После того, как FSYNC перешел в лог. 0, последовательные данные вдвигаются во входной регистр сдвига устройства по спадам уровня (от лог. 1 к лог. 0) сигнала SCLK для 16 тактовых импульсов. FSYNC может перейти в лог. 1 после 16-го спада SCLK, с соблюдением минимального времени между спадом SCLK и нарастанием FSYNC, время t_8 . Альтернативно FSYNC можно удерживать в лог. 0 на кратное 16 количество импульсов SCLK, и затем перевести FSYNC в лог. 1 после окончания передачи данных. Таким способом можно загрузить в микросхему непрерывный поток 16-битных слов, в то время как FSYNC удерживается в состоянии лог. 0; FSYNC перейдет в лог. 1 только после 16-того спада SCLK последнего загруженного слова.

Между отдельными операциями записи импульсы SCLK могут идти непрерывно, или они могут быть в любом состоянии - лог. 1 или лог. 0. В любом случае уровень на SCLK должен быть в лог. 1, когда сигнал FSYNC переводится в лог. 0 (t_{11}).

Пример того, как программировать AD9833, см. в апноуте [AN-1070](#) на сайте компании Analog Devices.

Питание AD9833. Алгоритм на рис. 26 показывает принцип работы с AD9833. Когда AD9833 включается подачей питания, она должна быть сброшена. Это сбрасывает внутренние регистры в 0, чтобы установить аналоговый выход в среднее состояние.

Чтобы избежать случайных сигналов на выходе DAC при инициализации AD9833, бит сброса должен быть установлен в 1, пока микросхема не будет готова генерировать выходной сигнал. Сброс не сбрасывает регистры фазы, частоты или управления. Эти регистры будут содержать недопустимые данные, и поэтому должны быть установлены пользователем в определенное значение. Бит сброса должен быть тогда установлен в 0, чтобы на выходе появился генерируемый сигнал. Данные появятся на выходе DAC через 7 или 8 тактов MCLK после того, как бит сброса будет установлен в 0.

Период латентности (задержка). Задержка здесь связана с асинхронной операцией записи в AD9833. Если в выбранный регистр частоты или фазы загружено новое слово, то имеется задержка 7 или 8 тактов MCLK перед тем, как поменяется сигнал на аналоговом выходе. Какая будет задержка - 7 или 8 циклов - зависит от позиции фронта нарастания MCLK, когда данные загружаются в регистр назначения.

Регистр управления. AD9833 содержит 16-битный регистр управления, который позволяет пользователю конфигурировать работу AD9833. Все биты управления, кроме бита режима (mode bit) анализируются по внутреннему спаду сигнала MCLK.

В таблице 6 описаны отдельные биты регистра управления. Различные функции и разные опции выхода AD9833 описаны более подробно в секции "Регистры частоты и фазы".

Чтобы оповестить AD9833, что содержимое регистра управления поменялось, биты D15 и D14 должны быть установлены в 0, как показано в таблице 5.

Таблица 5. Биты регистра управления.

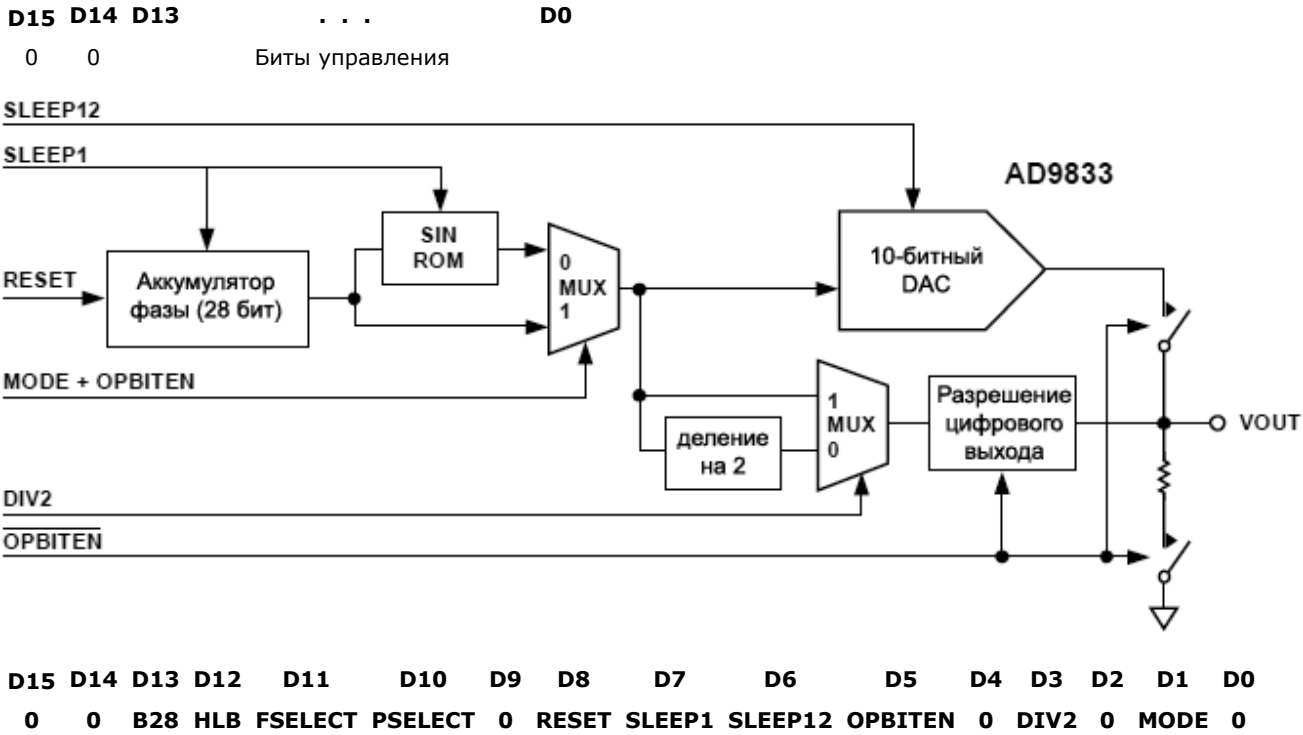


Рис. 24. Функции бит управления.

Таблица 6. Описание бит в регистре управления.

Бит	Имя	Функция
D13	B28	Для загрузки полного слова в любой из регистров частоты требуются 2 операции записи. B28 = 1 позволяет загрузить полное слово в регистр частоты двумя следующими друг за другом записями. Первая запись содержит 14 LSB (14 младших бит) слова частоты, и следующая запись содержит 14 MSB (14 старших бит). Первые 2 бита каждого 16-битного слова определяют тот регистр частоты, в который должно быть загружено слово, и таким образом, эти первые 2 бита должны быть одинаковыми для обоих следующих друг за другом записей. См. таблицу 8 для определения подходящего адреса. Запись в регистр частоты происходит после того, как будут загружены оба слова; таким образом, регистр никогда не получит промежуточную величину. Пример полной 28-битной записи показан в таблице 9. Когда B28 = 0, 28-битный регистр частоты работает как два 14-битных регистра, один из которых содержит 14 MSB, и другой 14 LSB. Это означает, что 14 MSB слова частоты может быть поменено независимо от 14 LSB, и наоборот. Чтобы поменять 14 MSB или 14 LSB делается одна запись по нужному адресу частоты. Управляющий бит D12 (HLB) информирует AD9833 какие должны быть изменены биты: 14 MSB или 14 LSB.
D12	HLB	Этот бит управления позволяет загрузить биты MSB или биты LSB регистра частоты, игнорируя при этом остальные 14 бит. Это полезно, если не требуется полная 28-битная разрешающая способность. Бит HLB используется вместе с битом D13 (B28). Этот управляющий бит показывает, куда попадут загружаемые 14 бит - в 14 MSB или 14 LSB адресованного регистра частоты. D13 (B28) должен быть установлен в 0, чтобы можно было отдельно поменять биты MSB и LSB слова

частоты. Когда D13 (B28) = 1, этот бит управления игнорируется. HLB = 1 позволяет записать 14 MSB адресованного регистра частоты, и соответственно HLB = 0 позволяет записать 14 LSB адресованного регистра частоты.

D11	FSELECT	Бит FSELECT определяет, какой из регистров используется в аккумуляторе фазы - FREQ0 или FREQ1.
D10	PSELECT	Бит PSELECT определяет, данные какого из регистров PHASE0 или PHASE1 добавляются к выходу аккумулятора фазы.
D9	Зарезервировано	Этот бит должен быть установлен в 0.
D8	Reset	Reset = 1 сбрасывает внутренние регистры в 0, что соответствует появлению на выходе среднего уровня. Reset = 0 запрещает сброс. Эта функция подробнее объяснена в таблице 13.
D7	SLEEP1	Когда SLEEP1 = 1, действие внутренней частоты MCLK запрещается, и выход DAC остается в своем текущем состоянии, потому что NCO больше не накапливает фазу. Когда SLEEP1 = 0, MCLK разрешается. Эта функция описана в таблице 14.
D6	SLEEP12	SLEEP12 = 1 выключает встроенный DAC (ЦАП). Это полезно, когда AD9833 используется в качестве вывода старшего бита данных DAC (формирование прямоугольного сигнала на выходе). SLEEP12 = 0 подразумевает, что DAC активен. Эта функция описана в таблице 14.
D5	OPBITEN	Функция этого бита вместо с битом D1 (mode, режим) управляет, что будет появляться на выходе VOUT. Это объяснено в таблице 15. Когда OPBITEN = 1, выход DAC не выводится на ножку VOUT, вместо этого частота смены состояния бита MSB (старший бит) данных DAC (или частота MSB/2) появляется на выходе VOUT. Это полезно использовать в качестве формирования тактовой частоты (грубого прямоугольного сигнала). Бит DIV2 управляет, подключен ли дополнительный делитель частоты MSB. Когда OPBITEN = 0, выход DAC подключается к VOUT. Бит mode (D1) определяет, какая будет форма сигнала на выходе - синусоидальная или треугольная.
D4	Зарезервировано	Этот бит должен быть установлен в 0.
D3	DIV2	DIV2 используется вместе с битом D5 (OPBITEN). Он описан подробнее в таблице 15. Когда DIV2 = 1, старший бит данных DAC (MSB) напрямую выводится на вывод VOUT. Когда DIV2 = 0, на выходе VOUT появляется прямоугольная частота MSB/2.
D2	Зарезервировано	Этот бит должен быть установлен в 0.
D1	Mode	Этот бит используется вместе с битом OPBITEN (D5). Он управляет, какой сигнал появляется на выходе VOUT, когда для формирования сигнала используется выход DAC. Этот бит должен быть установлен в 0, если управляющий бит OPBITEN = 1. Это объяснено в таблице 15. Когда mode = 1, SIN ROM пропускается, в результате на выходе DAC появляется треугольная форма сигнала. Когда mode = 0, SIN ROM используется для преобразования информации фазы в информацию амплитуды, в результате чего на выходе появляется синусоидальный сигнал.
D0	Зарезервировано	Этот бит должен быть установлен в 0.

Регистры частоты и фазы. AD9833 два регистра частоты и 2 регистра фазы, которые описаны в таблице 7.

Таблица 7. Регистры частоты и фазы.

Регистр Размер Описание

FREQ0	28 бит	Регистр частоты 0. Когда бит FSELECT=0, этот регистр определяет выходную частоту как фракцию частоты MCLK.
FREQ1	28 бит	Регистр частоты 1. Когда бит FSELECT=1, этот регистр определяет выходную частоту как фракцию частоты MCLK.
PHASE0	12 бит	Регистр смещения фазы 0. Когда бит PSELECT=0, содержимое этого регистра добавляется к выходу аккумулятора фазы.
PHASE1	12 бит	Регистр смещения фазы 1. Когда бит PSELECT=1, содержимое этого регистра добавляется к выходу аккумулятора фазы.

На аналоговом выходе AD9833 будет частота

$$f_{MCLK}/2^{28} * FREQREG$$

где FREQREG это значение, загруженное в выбранный регистр частоты. Этот сигнал по фазе сдвинут на

$$2\pi/4096 * PHASEREG$$

где PHASEREG это значение, содержащееся в выбранном регистре фазы. Должно быть уделено вниманию взаимосвязи между выбранной выходной частотой и опорной частотой, чтобы избежать нежелательных аномалий на выходе.

На рис. 28 показан алгоритм подпрограммы для записи регистров частоты и фазы AD9833.

Запись в регистр частоты

Когда записывается регистр частоты, биты D15 и D14 дают адрес регистра частоты.

Таблица 8. Биты регистра частоты.

D15	D14	D13	D0
0	1	Старшие 14 бит регистра FREQ0	LSB

1 0 Старшие 14 бит регистра FREQ1 LSB

Если пользователь хочет поменять все содержимое регистра частоты, необходимы 2 следующие друг за другом операции записи по одному и тому же адресу, потому что регистры частоты имеют разрядность 28 бит. Первая запись содержит 14 младших бит (LSB), и вторая запись содержит 14 старших бит (MSB). Для этого режима работы бит управления B28 (D13) должен быть установлен в 1. Пример 28-битной записи показан в таблице 9.

Таблица 9. Запись 0xFFFFC000 в регистр FREQ0.

Данные SDATA Результат ввода слова

0010 0000 0000 0000 Запись слова управления: (D15, D14) = 00, B28 (D13) = 1, HLB (D12) = X

0100 0000 0000 0000 Запись регистра FREQ0: (D15, D14) = 01, 14 бит LSB = 0x0000

0111 1111 1111 1111 Запись регистра FREQ0: (D15, D14) = 01, 14 бит MSB = 0x3FFF

В некоторых приложениях пользователю не нужно изменять все 28 бит регистра частоты. При грубой настройке нужно поменять только 14 MSB, в то время как при тонкой настройке требуется поменять только 14 LSB. Путем установки управляющего бита B28 (D13) в лог. 0, 28-битный регистр частоты работает как два 14-битных регистра, в одном из которых содержатся 14 MSB, и в другом содержатся 14 LSB. Это означает, что 14 бит MSB слова частоты можно поменять независимо от 14 бит LSB, и наоборот. Бит HLB (D12) в регистре управления идентифицирует, какие из этих 14 бит будут изменены. Примеры этого показаны в таблицах 10 и 11.

Таблица 10. Запись 0x3FFF в 14 младших бит (LSB) регистра FREQ1.

Данные SDATA Результат ввода слова

0000 0000 0000 0000 Запись слова управления: (D15, D14) = 00, B28 (D13) = 0, HLB (D12) = 0, поэтому будут записаны биты LSB

1011 1111 1111 1111 Запись регистра FREQ1: (D15, D14) = 10, 14 бит LSB = 0x3FFF

Таблица 11. Запись 0x00FF в 14 старших бит (MSB) регистра FREQ0.

Данные SDATA Результат ввода слова

0001 0000 0000 0000 Запись слова управления: (D15, D14) = 00, B28 (D13) = 0, HLB (D12) = 1, поэтому будут записаны биты MSB

0100 0000 1111 1111 Запись регистра FREQ0: (D15, D14) = 01, 14 бит MSB = 0x00FF

Запись в регистр фазы

Когда записывается регистр фазы, биты D15 и D14 устанавливаются в 11. Бит D13 идентифицирует, какой регистр фазы будет загружен.

Таблица 12. Биты регистра фазы.

D15	D14	D13	D12	D11	D0
1	1	0	X	12 старших бит (MSB) регистра PHASE0	LSB
1	1	1	X	12 старших бит (MSB) регистра PHASE1	LSB

Функция сброса. Эта функция сбросит внутренние регистры в 0, при этом аналоговый выход перейдет в среднее состояние. Сброс не сбрасывает регистры фазы, частоты или управления. Когда на микросхему AD9833 подается питание, она должна быть сброшена. Чтобы сбросить AD9833, установите бит сброса (reset bit) в лог. 1. Чтобы вывести микросхему из сброса, установите этот бит в лог. 0. Сигнал появится на выходе DAC через 8 тактов MCLK после того, как бит сброса будет установлен в 0.

Таблица 13. Применение функции сброса (влияние значения бита Reset).

Reset Результат

0	Сброс не прикладывается
1	Сброс внутренних регистров

Функция сна. Секции AD9833, которые не используются, могут быть выключены для минимизации потребления энергии. Это осуществляется функцией сна (sleep). Узлы микросхемы, которые могут быть отключены, это внутреннее тактирование и DAC. Биты, требуемые для задействования функции сна, описаны в таблице 14.

Таблица 14. Применение функции сна (влияние состояния битов SLEEP1 и SLEEP12).

SLEEP1 SLEEP12 Результат

0	0	Нет состояния Power Down, микросхема полностью активна
0	1	Выключен DAC
1	0	Запрещено внутреннее тактирование (MCLK не действует)

Выключение DAC

Это полезно, только когда AD9833 используется для вывода старшего разряда данных DAC. В таком случае DAC не нужен; таким образом, он может быть выключен для снижения энергопотребления.

Запрет внутреннего тактирования

Когда внутреннее тактирование AD9833 запрещено, выход DAC остается в своем текущем состоянии, потому что NCO больше не накапливает фазу. Новая частота, фаза и управляющее слово могут быть записаны в микросхему, когда активен бит управления SLEEP1. Такты синхронизации все еще остаются активными, это означает, что выбранные регистры частоты и фазы можно изменить с помощью бит управления. Установка бита SLEEP1 в лог. 0 разрешает тактирование MCLK. Любые изменения, сделанные в регистрах, когда SLEEP1 был активен (в лог. 1), будут видны на выходе после периода латентности.

Выход VOUT. AD9833 предоставляет различный вывод сигнала, все эти сигналы появляются на выводе VOUT. Выбранная форма сигнала определяется старшим битом (MSB) данных DAC: вывод синусоиды или вывод треугольника. Бит OPBITEN (D5) и бит режима (D1) регистра управления используются для выбора, какой вывод доступен из AD9833.

Старший бит данных DAC

MSB данных DAC может быть выведен из AD9833. Путем установки бита управления OPBITEN (D5) в лог. 1, старший бит данных DAC появляется на выводе VOUT. Это полезно в качестве грубого источника тактовой частоты. Частоту этого прямоугольного сигнала также можно поделить на 2 перед его выводом. Бит DIV2 (D3) регистра управления управляет частотой на выходе VOUT, будет она поделена на 2 или нет.

Генерация синусоидального сигнала

SIN ROM используется для преобразования информации о фазе из регистров частоты и фазы в информацию амплитуды, которая в результате дает на выходе синусоидальный сигнал. Чтобы получить синусоидальный сигнал на выходе VOUT, установите бит режима (D1) в лог. 0, и бит OPBITEN (D5) в лог. 0.

Генерация треугольного сигнала

SIN ROM может быть пропущено, и при этом обрезанный цифровой выход из NCO будет переправлен в DAC. Выходной сигнал больше не будет синусоидальным. DAC будет формировать 10-битную линейную треугольную функцию. Чтобы получить треугольную форму сигнала на выходе VOUT, установите бит режима (D1) в лог. 1.

Обратите внимание, что бит SLEEP12 должен быть установлен в 0 (т. е. работа DAC разрешена).

Таблица 15. Форма выходного сигнала на выводе VOUT (влияние состояния бит OPBITEN, Mode, DIV2).

OPBITEN	Mode	DIV2	Сигнал на выходе VOUT
0	0	X ¹	Синусоидальный сигнал
0	1	X ¹	Треугольный сигнал
1	0	0	На выходе появляется частота изменения старшего бита на входе DAC (MSB).
1	0	1	На выходе появляется частота изменения старшего бита на входе DAC, поделенная на 2 (MSB/2).
1	1	X ¹	Зарезервировано

Примечание 1: X = не имеет значения.



Рис. 25. Треугольный выходной сигнал.

[Информация по применению]

Из-за разных опций выходного сигнала AD9833 может быть сконфигурирована для широкого спектра приложений.

Одна из областей - приложения модуляции. Может быть реализована простая модуляция, такая как FSK (по русски это частотная манипуляция). Более сложные схемы модуляций, такие как GMSK (Гауссовская частотная модуляция с минимальным сдвигом) and QPSK (квадратурная фазовая манипуляция), также могут быть реализованы на AD9833.

В приложении FSK два регистра частоты загружаются разными значениями. Одна частота представляет частоту пробела (space frequency), в то время как другая представляет частоту маркера (mark frequency). С помощью программной манипуляции состоянием бита FSELECT в регистре управления пользователь может модулировать частоту несущей этими двумя значениями.

У AD9833 есть 2 регистра фазы, которые позволяют реализовать модуляцию PSK. При манипулировании сдвигом фазы получается модуляция PSK, при этом фаза меняется на величину, связанную с битовым потоком, поступающим на вход модулятора.

AD9833 также подходит для приложений генератора сигналов. Из-за того, что MSB данных DAC доступен на выводе VOUT, устройство может быть использовано для генерации прямоугольного сигнала.

С малым энергопотреблением микросхема подходит для приложений, где она может использоваться как локальный генератор в переносных устройствах.

Заземление и разводка печатной платы. Печатная плата (printed circuit board, PCB), на которой размещена AD9833, должна быть разработана так, чтобы аналоговая и цифровая секции цепей были разделены и ограничены определенными областями платы. Это упрощает использование отдельных заливок шиной земли (ground plane). Техника разводки с минимальным трением меди обычно лучше всего подходит для заливок, потому что дает лучшее экранирование. Цифровые и аналоговые заливки должны быть соединены друг с другом только в одном месте. Если AD9833 является единственным устройством, требующим соединения AGND с DGND, то их заливки должны подключаться к цепям AGND и DGND на выводах AD9833. Если AD9833 находится в системе, где несколько устройств требуют соединений AGND с DGND, то соединение должно быть сделано в одной точке, сформированной в виде звезды, которая должна находиться как можно ближе к AD9833.

Избегайте прокладки цифровых сигналов под микросхемой AD9833, поскольку они вызывают наводки на аналоговые сигналы. Аналоговой заливке следует позволить проходить под AD9833, чтобы избежать наводок шума. Линии питания, приходящие к AD9833, должны использовать по возможности широкие проводники, чтобы уменьшить сопротивление цепей тока и снизить эффекты пульсаций на линии питания. Быстро переключающиеся сигналы, такие как такты, должны быть экранированы цифровой землей, чтобы избежать наводок шума на другие секции платы.

Избегайте пересечения аналоговых и цифровых сигналов. Проводники на противоположных сторонах платы должны находиться друг к другу под прямым углом. Это снижает эффекты перекрестных помех между сигналами через плату. Микрополосковая техника является безусловно лучшей, но она не всегда возможна на двусторонней плате. В такой технике сторона компонентов выделена целиком (или почти целиком) под заливки медью, и все сигналы стараются разместить на противоположной стороне.

Важна хорошая блокировка конденсаторами цепей питания. К AD9833 должны быть подключены блокировочные керамические конденсаторы 0.1 мкф параллельно танталовым конденсаторам на 10 мкф. Чтобы достичь наибольшей эффективности развязывающих конденсаторов, они должны быть размещены максимально близко к корпусу микросхемы.

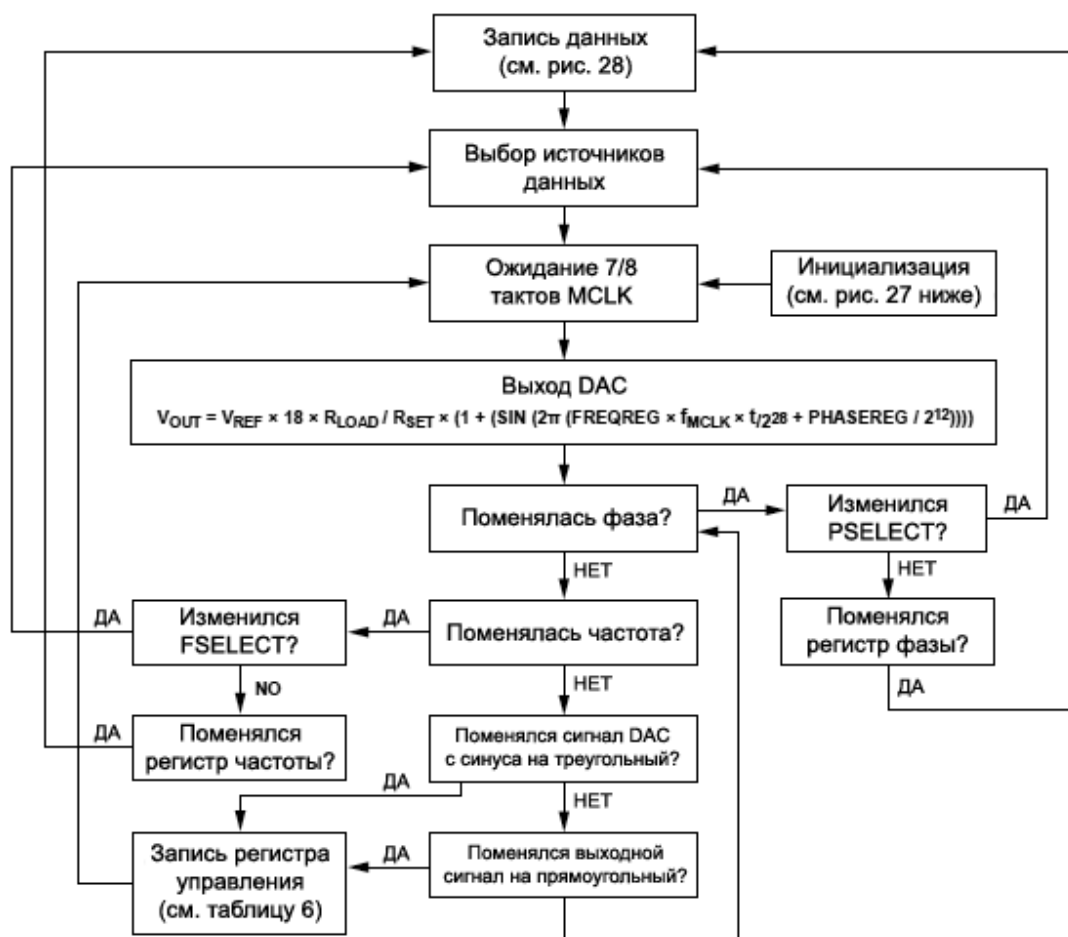


Рис. 26. Алгоритм инициализации и работы AD9833.

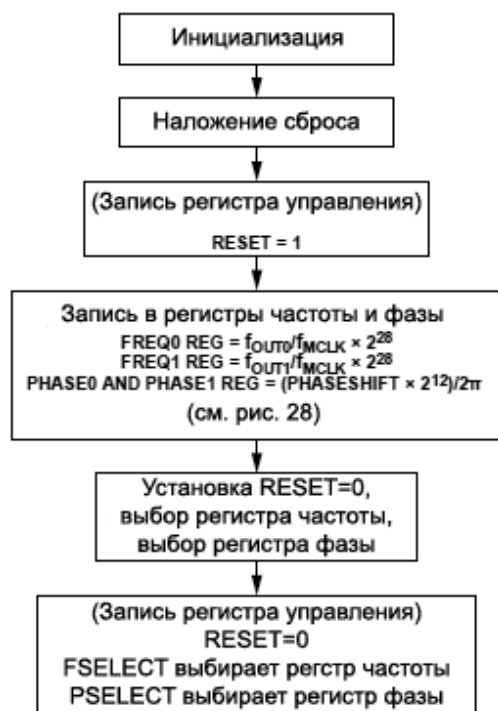


Рис. 27. Алгоритм инициализации.

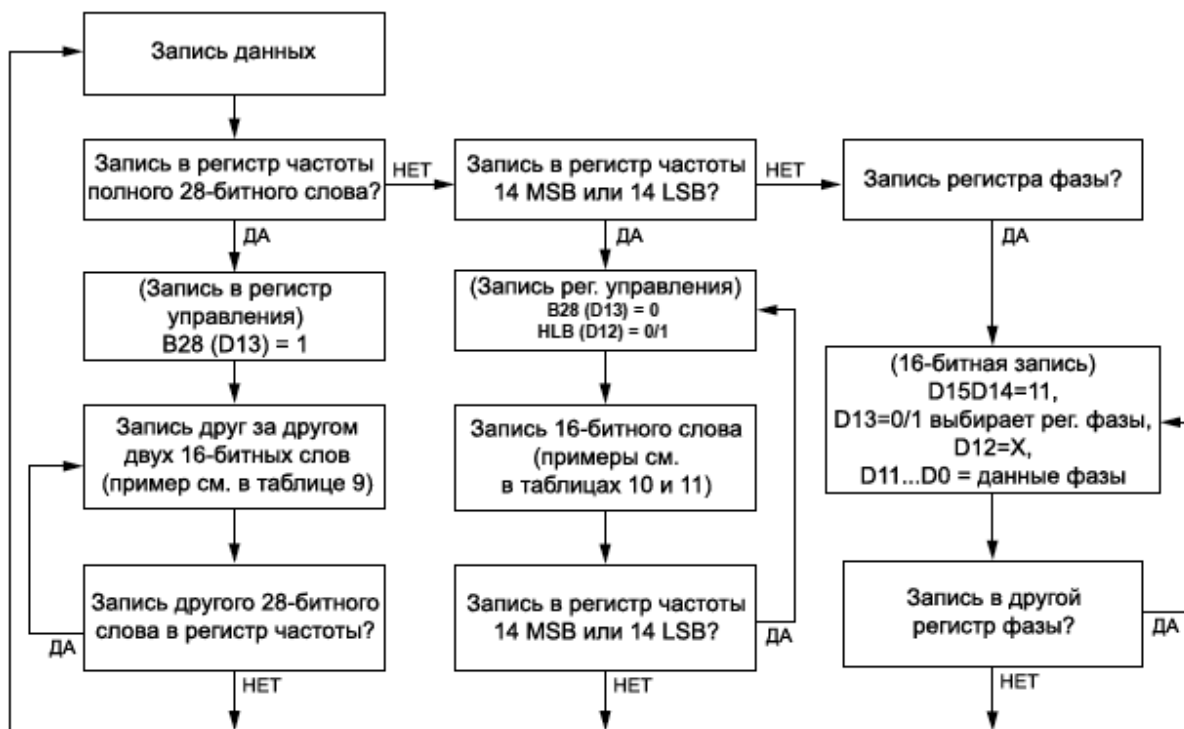


Рис. 28. Алгоритм операций записи данных.

[Подключение к микроконтроллеру]

AD9833 снабжена стандартным последовательным интерфейсом, что позволяет подключить её напрямую к любому микроконтроллеру. Устройство использует внешнее тактирование для записи данных или информации управления. Такты для данных могут быть с максимальной частотой 40 МГц. Такты данных могут следовать непрерывно, или могут иметь статический уровень (0 или 1) в неактивном состоянии между операциями записи. Когда данные или управляющая информация записываются в AD9833, сигнал FSYNC переводится в лог. 0 и удерживается в таком состоянии, пока в AD9833 не будут записаны 16 данных. Сигнал FSYNC задает фреймы информации по 16 бит, загружаемые в AD9833.

Интерфейс 68HC11/68L11. На рис. 29 показан последовательный интерфейс между AD9833 и микроконтроллером 68HC11/68L11. Микроконтроллер сконфигурирован как мастер установкой в лог. 1 бита MSTR регистра SPCR. Эта установка предоставляет тактовый сигнал данных на вывод SCK; выход MOSI поставляет последовательные данные сигнала SDATA. Поскольку у микроконтроллера нет специального выделенного вывода для сигнала синхронизации фрейма, то сигнал FSYNC получается от ножки порта GPIO (PC7). Условия настройки для корректной работы интерфейса следующие:

- SCK удерживается в лог. 1 между операциями записи (CPOL = 0)
- Данные MOSI достоверны на спаде сигнала SCK (CPHA = 1)

Когда данные передаются в AD9833, сигнал FSYNC переводится в лог. 0 (ножкой порта PC7). Последовательные данные из 68HC11/68L11 передаются по байтам (2 посылки по 8 бит), 8 перепадов тактов происходят в одном цикле передачи. Данные передаются старшим битом (MSB) вперед. Для загрузки данных в AD9833 ножка PC7 удерживается в лог. 0 после передачи 8 бит, и выполняется вторая операция записи в AD9833. Только после второй посылки из 8 бит сигнал FSYNC должен быть переведен в лог. 1. Таким способом формируется фрейм из 16 бит данных.

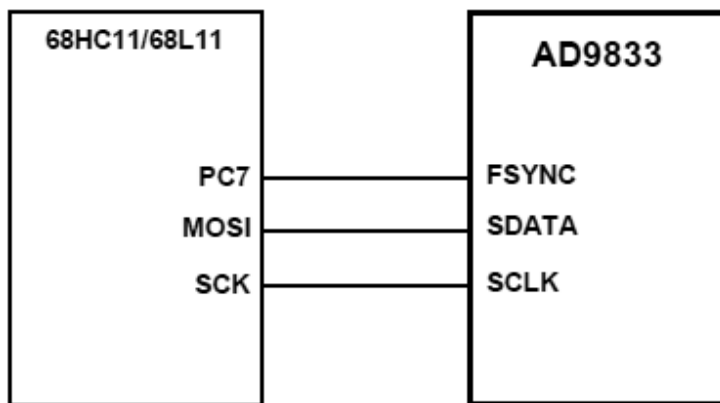


Рис. 29. Подключение 68HC11/68L11 к интерфейсу AD9833.

Интерфейс 80C51/80L51. На рис. 30 показан последовательный интерфейс между AD9833 и микроконтроллером 80C51/80L51. Микроконтроллер работает в Mode 0, так что TxD чипа 80C51/80L51 управляет сигналом SCLK AD9833, и ножка RxD управляет сигналом данных SDATA. Сигнал FSYNC получается от ножки GPIO P3.3.

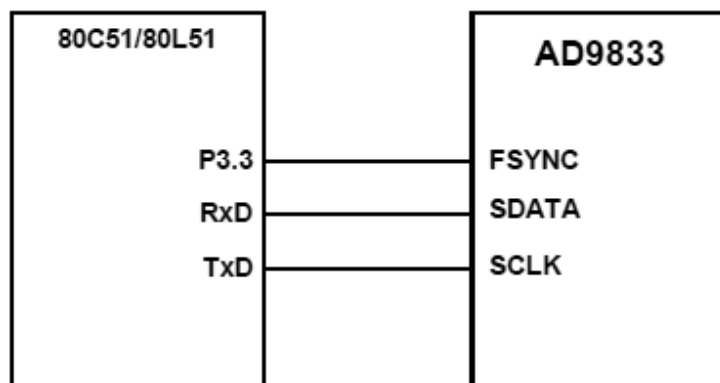


Рис. 30. Подключение 80C51/80L51 к интерфейсу AD9833.

Когда данные передаются в AD9833, порт P3.3 удерживается в лог. 0. 80C51/80L51 передает данные в по 8 бит, так что только 8 перепадов тактов появляются в каждом цикле. Чтобы загрузить оставшуюся порцию из 8 бит в AD9833, порт P3.3 удерживается в лог. 0 после передачи первой порции из 8 бит, и инициируется вторая операция записи для передачи второго байта данных. P3.3 переводится в лог. 1 после завершения второй операции записи. Сигнал SCLK должен удерживаться в лог. 1 между двумя операциями записи.

80C51/80L51 выводит последовательные данные в формате, когда LSB идет первым. Но AD9833 принимает первым бит MSB (первые 4 MSB содержат информацию управления, следующие 4 бита содержат адрес, и оставшиеся 8 бит LSB содержат данные, когда они записываются в регистр назначения). Таким образом, подпрограмма передачи должна учитывать это обстоятельство, чтобы перетасовать порядок следования бит так, чтобы MSB выводился первым.

Интерфейс DSP56002. Рис. 31 показывает интерфейс между AD9833 и DSP56002. DSP56002 конфигурируется для нормального асинхронного режима работы с управляемым внутренним тактированием (SYN = 0, GCK = 1, SCKD = 1). Сигнал фрейма генерируется внутри микроконтроллера (SC2 = 1), передачи настраиваются на длину 16 бит (WL1 = 1, WL0 = 0), и сигнал синхронизации фрейма соответствует 16 битам (FSL = 0). Сигнал синхронизации фрейма доступен на выводе SC2, но он должен быть проинвертирован перед тем, как подать его на AD9833. Интерфейс с DSP56000/DSP56001 подобен интерфейсу с DSP56002.

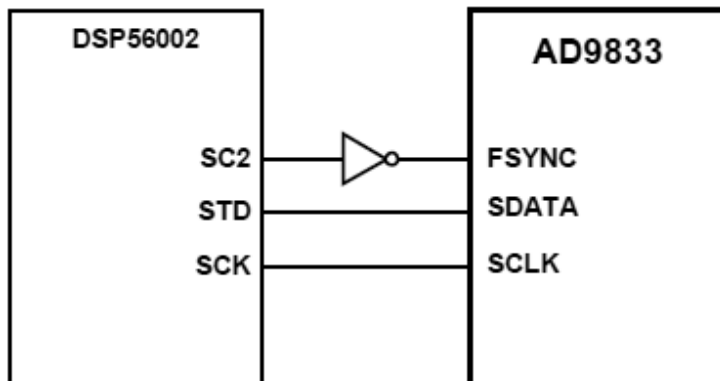
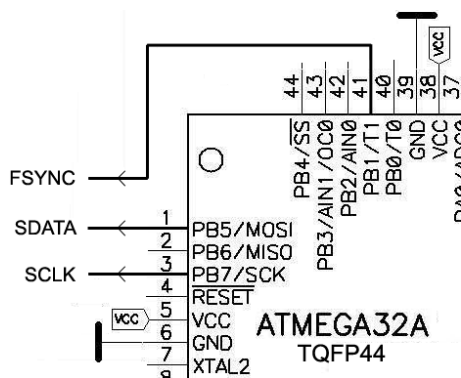


Рис. 31. Подключение DSP56002 к интерфейсу AD9833.

Интерфейс с AVR. На рисунке ниже показан пример подключения AD9833 к микроконтроллеру ATmega32A. В качестве канала связи используется интерфейс SPI, сигнал FSYNC управляется аппаратно, с помощью порта GPIO PB1.



Во врезке приведен пример программирования.

Подпрограммы управления AD9833 для микроконтроллера ATmega32A

[ad9833.h]

```

/*
 * File: code for controlling the ad9833 DDS
 * Author: Tuomas Nylund (tuomas.nylund@gmail.com)
 * Website: http://tuomasnylund.fi*
 * This program is free software: you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program. If not, see http://www.gnu.org/licenses/.
 */

/*
 *
 * \file ad9833.h
 *
 * Заголовочный файл для ad9833.c.
 */

/**\defgroup ad9833_h ad9833.h - Библиотека AVR для AD9833
 * Эта библиотека работает с чипом DDS AD9833. Для неё нужна
 * библиотека \ref spi_h
 */
/* \section ad9833_example Example
 * \code
 * // Инициализация AD9833:
 * ad9833_init();
 *
 * // Установка регистра частоты 0 на частоту 10 кГц:
 * ad9833_set_frequency(0,10000);
 *
 * // Запуск генерации из регистра частоты 0:
 * ad9833_set_freq_out(0);
 *
 * // Установка синусоидальной формы сигнала:
 * ad9833_set_mode(AD_SINE);
 * \endcode
 */
/*
 *
 * \ingroup ad9833
 * @{
 */

#ifndef _AD9833_H_
#define _AD9833_H_

/** \name AD9833 Настройки и макросы для управления выводом FrameSync
 * @{
 */
#define AD_FSYNC_PORT PORTB

```

```

#define AD_FSYNC_DDR    DDRB
#define AD_FSYNC_BIT    PB1

#define AD_FSYNC_HI() AD_FSYNC_PORT |= (1 << AD_FSYNC_BIT)
#define AD_FSYNC_LO() AD_FSYNC_PORT &= ~(1 << AD_FSYNC_BIT)
/** @} */

/** \name Режимы вывода формы сигнала AD9833
 * Параметры для \ref ad9833_set_mode()
 * @{
 */
#define AD_OFF          0
#define AD_TRIANGLE     1
#define AD_SQUARE       2
#define AD_SINE         3
/** @} */

/** \name Биты регистра команд AD9833
 * @{
 */
#define AD_B28          13
#define AD_HLB          12
#define AD_FSELECT      11
#define AD_PSELECT      10
#define AD_RESET        8
#define AD_SLEEP1       7
#define AD_SLEEP12      6
#define AD_OPBITEN      5
#define AD_DIV2          3
#define AD_MODE          1
/** @} */

/** \name Адреса регистров AD9833
 * @{
 */
#define AD_FREQ0 (1 << 14)
#define AD_FREQ1 (1 << 15)
#define AD_PHASE0 (3 << 14)
#define AD_PHASE1 ((3 << 14)|(1 << 13))
/** @} */

/** \name Макросы для вычислений AD9833
 * @{
 */
#define AD_F_MCLK 25000000 // Опорная частота тактов ad9833
#define AD_2POW28 268435456 // Используется для вычисления выходной частоты

/** Макрос, который вычисляет значение регистра частоты по заданной частоте */
#define AD_FREQ_CALC(freq) (uint32_t)(((double)AD_2POW28/(double)AD_F_MCLK*freq)*4)

/** Макрос, который вычисляет значение для прерывания
 * Timer1 output compare по заданной частоте модуляции */
#define AD_MOD_FREQ_CALC(freq) (F_CPU/(64*(uint32_t)freq))

/** Макрос, который вычисляет значение регистра фазы ad9833
 * по значению фазы в градусах */
#define AD_PHASE_CALC(phase_deg) (uint16_t)((512*phase_deg)/45)
/** @} */

/** \name Макросы для Timer1
 * @{
 */
#define TIMER_START() ICCR1B |= (1 << CS11)|(1 << CS10)
#define TIMER_STOP()  ICCR1B &= ~((1 << CS11)|(1 << CS10))
/** @} */

/** Структура, где хранятся все значения конфигурации, инициализированные
 * в глобальной переменной файла ad9833.c
 */
typedef struct {
    float    freq[2]; // Здесь хранятся частоты
    float    phase[2];
    float    mod_freq;
    uint8_t  freq_out;
    uint8_t  phase_out;
    uint8_t  mode;
    uint16_t command_reg;
} ad9833_settings_t;

```

```

// Некоторые функции возможно следует реализовать встраиваемыми (inline)
// для получения большей эффективности...

void ad9833_init(void);

void ad9833_set_mode(uint8_t mode);

void ad9833_set_frequency(uint8_t reg, double freq);
double ad9833_get_frequency(uint8_t reg);

void ad9833_set_phase(uint8_t reg, double phase);
double ad9833_get_phase(uint8_t reg);

void ad9833_set_freq_out(uint8_t freq_out);
uint8_t ad9833_get_freq_out(void);

void ad9833_set_phase_out(uint8_t phase_out);
uint8_t ad9833_get_phase_out(void);

void ad9833_set_mod_freq(uint16_t freq);
void ad9833_set_mod_bytes(uint8_t num, uint8_t *bytes, uint8_t repeat);

#endif
/** @} */

```

[ad9833.c]

```

/*
 * File: code for controlling the ad9833 DDS
 * Author: Tuomas Nylund (tuomas.nylund@gmail.com)
 * Website: http://tuomasnylund.fi
 */

/**
 * \file ad9833.c
 *
 * Функции для управления чипом AD9833 DDS через микроконтроллер AVR.
 * Этот чип позволяет генерить синус, прямоугольник и треугольник на
 * частотах от примерно 0.1 Гц до 3 МГц.
 *
 * Он также позволяет осуществлять простую двоичную модуляцию в форме
 * FSK и PSK. Также должна быть возможна амплитудная модуляция, но она
 * здесь пока не реализована.
 */
#include <avr/io.h>
#include <avr/interrupt.h>
#include <util/delay.h>
#include "ad9833.h"
#include "spi.h"

ad9833_settings_t ad_settings; // Это используется для сохранения настроек.

/**
 * Функция обертки для отправки 16-битных пакетов SPI.
 * \param packet 16-битное значение для отправки через SPI.
 */
static inline void ad9833_send(uint16_t packet)
{
    spi_send_byte((uint8_t)(packet >> 8));
    spi_send_byte((uint8_t)packet);
}

/**
 * Инициализирует AD9833 и относящиеся к ней переменные.
 * Также инициализируется Timer1, используемый для для
 * реализации модуляции.
 */
void ad9833_init(void)
{
    //Инициализация вывода FSYNC (он работает как выборка
    // с активным лог. 0):
    ad_settings.command_reg |= (1 << AD_B28);
    AD_FSYNC_DDR |= (1 << AD_FSYNC_BIT);
    AD_FSYNC_HI();
}

```

```

//Инициализация модулирующего таймера:
TCCR1B |= (1 << WGM12); // Таймер в режиме CTC
TCCR1B |= (1 << CS11)|(1 << CS10); // Делитель тактов на 64
TIMSK |= (1 << OCIE1A);
OCR1A = 0xFFFF0;

// Задержка, взятая из даташита:
_delay_us(10);

// Начало с выключением питания DAC и сбросом
AD_FSYNC_LO();
_delay_us(5);
ad9833_send((1 << AD_SLEEP12)|(1 << AD_RESET));
ad_settings.command_reg |= (1 << AD_SLEEP12);
_delay_us(5);
AD_FSYNC_HI();

// Установка некоторых хороших значений по умолчанию.
ad9833_set_frequency(0, 0);
ad9833_set_frequency(1, 0);
ad9833_set_phase(0, 0);
ad9833_set_phase(1, 0);
ad9833_set_freq_out(0);
ad9833_set_phase_out(0);
}

/**
 * Установка формы выходного сигнала ad9833 в соответствии с входным параметром.
 * \param mode возможные значения:
 * - AD_OFF
 * - AD_TRIANGLE
 * - AD_SQUARE
 * - AD_SINE
 */
void ad9833_set_mode(uint8_t mode){
    ad_settings.mode = mode;
    switch (mode){
        case AD_OFF:
            ad_settings.command_reg |= (1 << AD_SLEEP12);
            ad_settings.command_reg |= (1 << AD_SLEEP1);
            break;
        case AD_TRIANGLE:
            ad_settings.command_reg &= ~(1 << AD_OPBITEN);
            ad_settings.command_reg |= (1 << AD_MODE);
            ad_settings.command_reg &= ~(1 << AD_SLEEP12);
            ad_settings.command_reg &= ~(1 << AD_SLEEP1);
            break;
        case AD_SQUARE:
            ad_settings.command_reg |= (1 << AD_OPBITEN);
            ad_settings.command_reg &= ~(1 << AD_MODE);
            ad_settings.command_reg |= (1 << AD_DIV2);
            ad_settings.command_reg &= ~(1 << AD_SLEEP12);
            ad_settings.command_reg &= ~(1 << AD_SLEEP1);
            break;
        case AD_SINE:
            ad_settings.command_reg &= ~(1 << AD_OPBITEN);
            ad_settings.command_reg &= ~(1 << AD_MODE);
            ad_settings.command_reg &= ~(1 << AD_SLEEP12);
            ad_settings.command_reg &= ~(1 << AD_SLEEP1);
            break;
    }
    AD_FSYNC_LO();
    // _delay_us(5);
    ad9833_send(ad_settings.command_reg);
    // _delay_us(5);
    AD_FSYNC_HI();
}

/**
 * Установка требуемого внутреннего регистра фазы ad9833.
 *
 * \param reg номер регистра фазы, 0 или 1
 * \param phase требуемая фаза
 */
void ad9833_set_phase(uint8_t reg, double phase){
    uint16_t reg_reg;
    if (reg==1)
        reg_reg = AD_PHASE1;

```

```

else
    reg_reg = AD_PHASE0;

ad_settings.phase[reg] = phase;

AD_FSYNC_LO();
//_delay_us(5);
ad9833_send(reg_reg | AD_PHASE_CALC(ad_settings.phase[reg]));
//_delay_us(5);
AD_FSYNC_HI();
}

/**
 * Возвращает фазу выбранного регистра
 * \param reg регистр, фазу которого нужно получить
 * \return фаза выбранного регистра
 */
double ad9833_get_phase(uint8_t reg){
    return ad_settings.phase[reg];
}

/**
 * Выбирает регистр частоты, который используется для генерации
 * выходного сигнала. Также используется для модуляции
 * FSK (частотная манипуляция).
 * \param freq_out возможные значения:
 * - 0 = использовать регистр частоты 0
 * - 1 = использовать регистр частоты 1
 * - 2 = FSK
 */
void ad9833_set_freq_out(uint8_t freq_out){
    ad_settings.freq_out = freq_out;
    switch (freq_out){
        case 0:
            ad_settings.command_reg &= ~(1 << AD_FSELECT);
            break;
        case 1:
            ad_settings.command_reg |= (1 << AD_FSELECT);
            break;
        case 2:
            // Пока не реализовано...
            break;
    }

    AD_FSYNC_LO();
    //_delay_us(5);
    ad9833_send(ad_settings.command_reg);
    //_delay_us(5);
    AD_FSYNC_HI();
}

/**
 * Возвращает ранее установленный номер регистра
 * вывода частоты.
 * \return ранее установленный регистр частоты
 */
uint8_t ad9833_get_freq_out(void){
    return ad_settings.freq_out;
}

/**
 * Выбирает регистр фазы, используемый для генерации
 * выходного сигнала. Также используется для
 * модуляции PSK (фазовая манипуляция).
 * \param phase_out возможные значения:
 * - 0 = использовать регистр фазы 0
 * - 1 = использовать регистр фазы 1
 * - 2 = PSK
 */
void ad9833_set_phase_out(uint8_t phase_out){
    ad_settings.phase_out = phase_out;
    switch (phase_out){
        case 0:
            ad_settings.command_reg &= ~(1 << AD_PSELECT);
            break;
        case 1:
            ad_settings.command_reg |= (1 << AD_PSELECT);
            break;
        case 2:

```



```

        // Пока не реализовано...
        break;
    }

    AD_FSYNC_LO();
    // _delay_us(5);
    ad9833_send(ad_settings.command_reg);
    // _delay_us(5);
    AD_FSYNC_HI();
}

/**
 * Вернет ранее установленный режим выхода фазы.
 * \return the ранее установленный режим фазы
 */
uint8_t ad9833_get_phase_out(void){
    return ad_settings.phase_out;
}

/**
 * Устанавливает внутренний регистр частоты ad9833 на значение,
 * соответствующее выбранной частоте.
 *
 * \param reg регистр частоты, 0 или 1
 * \param freq требуемая частота
 */
void ad9833_set_frequency(uint8_t reg, double freq){
    uint32_t freq_reg;
    uint16_t reg_reg;
    freq_reg = AD_FREQ_CALC(freq);
    ad_settings.freq[reg] = freq;

    if (reg==1)
        reg_reg = AD_FREQ1;
    else
        reg_reg = AD_FREQ0;

    AD_FSYNC_LO();
    // _delay_us(5);
    ad9833_send((1 << AD_B28) | ad_settings.command_reg);
    ad9833_send(reg_reg | (0x3FFF & (uint16_t)(freq_reg >> 2)));
    ad9833_send(reg_reg | (0x3FFF & (uint16_t)(freq_reg >> 16)));
    // _delay_us(5);
    AD_FSYNC_HI();
}

/**
 * Возвратит выбранный регистр.
 * \param reg регистр, значение которого хотим получить
 * \return частота выбранного регистра
 */
double ad9833_get_frequency(uint8_t reg){
    return ad_settings.freq[reg];
}

/**
 * Устанавливает частоту модуляции.
 *
 * \param freq выбранная частота модуляции
 */
void ad9833_set_mod_freq(uint16_t freq){
    ad_settings.mod_freq = freq;
    OCR1A = AD_MOD_FREQ_CALC(freq);
}

/** * Устанавливает байты для модуляции
 * ПОКА НЕ РЕАЛИЗОВАНО
 *
 * \param num количество отправляемых байт
 * \param bytes указатель на массив отправляемых байт
 * \param repeat сколько раз повторить отправку
 */
void ad9833_set_mod_bytes(uint8_t num, uint8_t *bytes, uint8_t repeat){
    //TODO ...
}

/**
 * Прерывание таймера, обрабатывающее модуляцию

```

```

*/
ISR(TIMER1_COMPA_vect){
    uint16_t check = ad_settings.command_reg;
    //TODO: реализовать модуляцию для реальных сигналов
    if (ad_settings.freq_out == 2)
        ad_settings.command_reg ^= ((uint16_t)1 << AD_FSELECT);
    if (ad_settings.phase_out == 2)
        ad_settings.command_reg ^= ((uint16_t)1 << AD_PSELECT);

    if (check != ad_settings.command_reg){
        AD_FSYNC_LO();
        //_delay_us(5);
        ad9833_send(ad_settings.command_reg);
        //_delay_us(5);
        AD_FSYNC_HI();
    }
}
}

```

[spi.h]

```

/** File: simple SPI routines for AVR
 * Author: Tuomas Nylund (tuomas.nylund@gmail.com)
 * Website: http://tuomasnylund.fi
 *
 * This program is free software: you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program. If not, see http://www.gnu.org/licenses/.
 */

/** \file spi.h
 * Заголовочный файл для spi.c
 */
#ifndef SPI_H_
#define SPI_H_

/** \defgroup spi_h spi.h - Библиотека AVR SPI.
 * Это очень простая библиотека SPI, которая использует внутреннее
 * периферийное устройство SPI микроконтроллера AVR.
 *
 * Она не использует прерывания, вместо этого применен опрос флага.
 *
 * Любые сигналы выборки можно реализовать самостоятельно с помощью
 * манипуляции выводами GPIO микроконтроллера.
 *
 * \section example Example
 * \code
 * // Инициализация SPI:
 * spi_init(SPI_MODE0|MSB_FIRST|CLOCKDIV4);
 *
 * // Отправка байта:
 * spi_send_byte(0xAA);
 *
 * // Прием байта:
 * uint8_t rcv;
 * rcv = spi_read_byte();
 * \endcode
 *
 * @{ */

#include <avr/io.h>

/** \name Определение ножек SPI
 * @{ */
#define SPI_DDR      DDRB
#define SPI_MOSI_BIT PB5
#define SPI_SCK_BIT  PB7
#define SPI_CS_BIT   PB4

```

```

/** @} */

/** \name Маски режима SPI
 * используются в \ref spi_init()
 * @{ */
#define SPIMODE0    (0 << CPHA)
#define SPIMODE1    (1 << CPHA)
#define SPIMODE2    (2 << CPHA)
#define SPIMODE3    (3 << CPHA)
/** @} */

/** \name Маски, управляющие порядком следования бит данных SPI
 * используются в \ref spi_init()
 * @{ */
#define LSB_FIRST    (1 << DORD)
#define MSB_FIRST    (0 << DORD)
/** @} */

/** \name Маски управления скоростью тактов SPI
 * используются в \ref spi_init()
 * @{ */
#define CLOCKDIV4    (0 << SPR0)
#define CLOCKDIV16   (1 << SPR0)
#define CLOCKDIV64   (2 << SPR0)
/** @} */

void spi_init(uint8_t setup);

void spi_send_byte(uint8_t data);

uint8_t spi_read_byte(void);

#endif
/** @} */

```

[spi.c]

```

/** File: simple SPI routines for AVR
 * Author: Tuomas Nylund (tuomas.nylund@gmail.com)
 * Website: http://tuomasnylund.fi
 */

/** \file spi.c
 * Простой код для использования периферийного устройства AVR SPI,
 * работающий по опросу (polling, т. е. без прерываний)
 */
#include "spi.h"

/**
 * Инициализирует AVR SPI в соответствии с указанными параметрами.
 * \param setup это опции SPI, которые могут быть маской, составленной
 * из SPIMODE*, *_FIRST, CLOCKDIV*. */
void spi_init(uint8_t setup)
{
    SPI_DDR |= (1 << SPI_MOSI_BIT)|(1 << SPI_SCK_BIT)|(1 << SPI_CS_BIT);
    SPCR     = (1 << SPE)|(1 << MSTR)|setup;
}

/**
 * Посылает один байт через SPI с использованием опроса.
 * \param data байт для отправки
 */
void spi_send_byte(uint8_t data)
{
    SPDR = data;
    while(!(SPSR & (1 << SPIF)));
}

/**
 * Прием одного байта через SPI с использованием опроса.
 * \return принятый байт
 */
uint8_t spi_read_byte(void)
{
    SPDR = 0x00;
    while(!(SPSR & (1 << SPIF)));
}

```

```
    return SPDR;  
}
```

Пример использования:

```
/* Инициализация SPI */  
spi_init(SPI_MODE2);  
  
/* Инициализация AD9833 */  
ad9833_init();  
  
/* Программирование формы сигнала wave и частоты freq */  
ad9833_set_frequency(0, freq);  
switch(wave)  
{  
case SYM_SINE:  
    ad9833_set_mode(AD_SINE);  
    mcp41010_write(levelSINE);  
    break;  
case SYM_TRIA:  
    ad9833_set_mode(AD_TRIANGLE);  
    mcp41010_write(levelTRIANGLE);  
    break;  
case SYM_SQUA:  
    ad9833_set_mode(AD_SQUARE);  
    break;  
}
```

Оценочная плата AD9833

Оценочная плата (AD9833 evaluation board) позволяет разработчикам попробовать в работе высокопроизводительный модулятор AD9833 DDS с минимальными затратами усилий.

[Платформа демонстрации]

Системная платформа разработки (System demonstration platform, SDP) это аппаратура и оценочное инструментальное программное обеспечение, которое используется вместе с оценочными платами разработчика. Плата SDP основана на процессоре **Blackfin® ADSP-BF527** с USB-подключением к PC через высокоскоростной порт USB 2.0.

Имейте в виду, что плата SDP продается отдельно от оценочной платы AD9833.

[Подключение AD9833 через SPORT]

Плата SDP компании Analog Devices имеет последовательный порт SPORT, который используется для управления входами последовательного интерфейса AD9833. Соединения показаны на рис. 32.

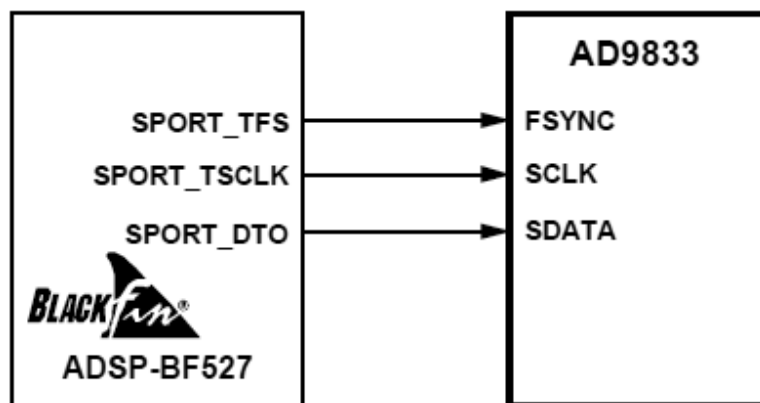


Рис. 32. Подключение SDP к интерфейсу AD9833.

[EVALUATION KIT]

Оценочный кит (DDS evaluation kit) включает готовую, проверенную печатную плату с AD9833. Схема оценочной платы показана на рис. 34 и 35.

Программное обеспечение, поставляемое вместе с оценочным китом, позволяет пользователю просто программировать AD9833 (см. рис. 33). Программное обеспечение запускается на любом IBM-совместимом PC с установленной системой Microsoft® Windows® (включая Windows 7). Программа совместима как с 32-битной, так и с 64-битной операционными системами.



Рис. 33. Графический интерфейс AD9833 Evaluation Software.

Больше информации об оценочном программном обеспечении можно получить на CD и на [страничке продукта AD9833](#).

[Кварцевые генераторы против внешнего тактирования]

AD9833 может работать с основной опорной частотой до 25 МГц. На оценочной плате имеется установленный генератор 25 МГц. Этот генератор может быть удален, и если это требуется, то к микросхеме может быть подключен внешний тактовый сигнал CMOS. Как вариант, могут быть следующие варианты обычного генератора:

- Генераторы AEL 301-Series, компания AEL Crystals
- Генераторы SG-310SCN, компания Epson Electronics

[Питание]

Питание для оценочной платы AD9833 может быть предоставлено через коннектор USB или подано снаружи через подключения к выводам. Провода питания должны быть повиты друг с другом, чтобы уменьшить помехи от наводок.

[Принципиальные схемы]

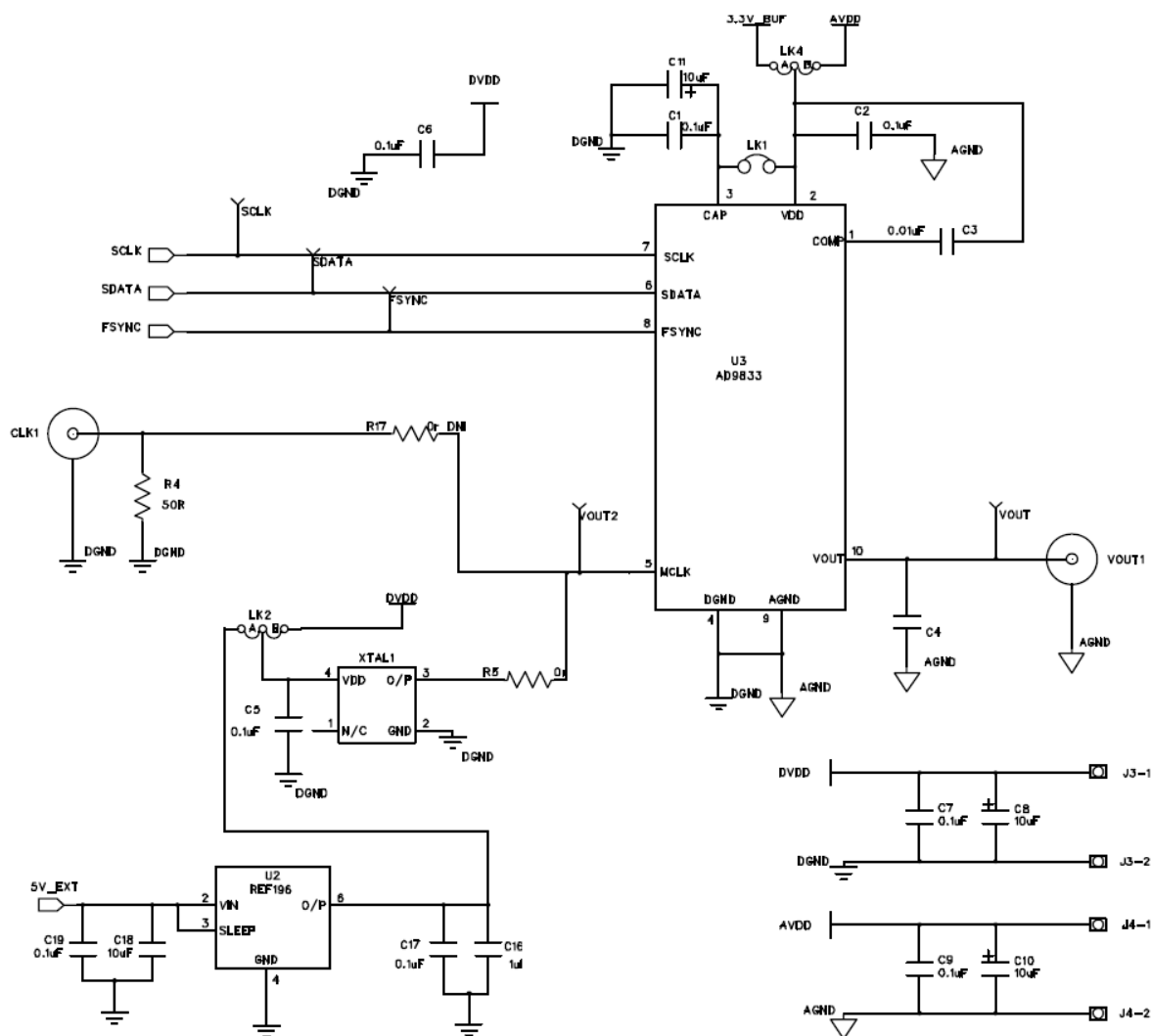


Рис. 34. Схема оценочной платы AD9833.

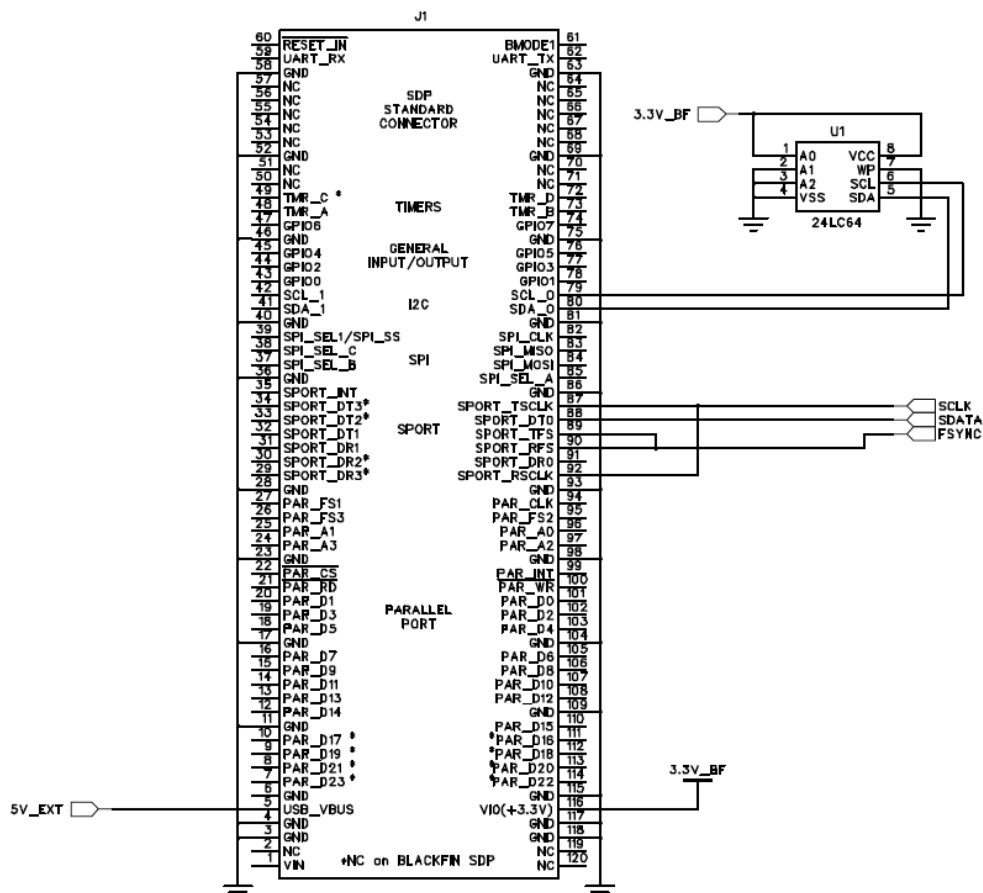


Рис. 35. Схема коннектора SDP.

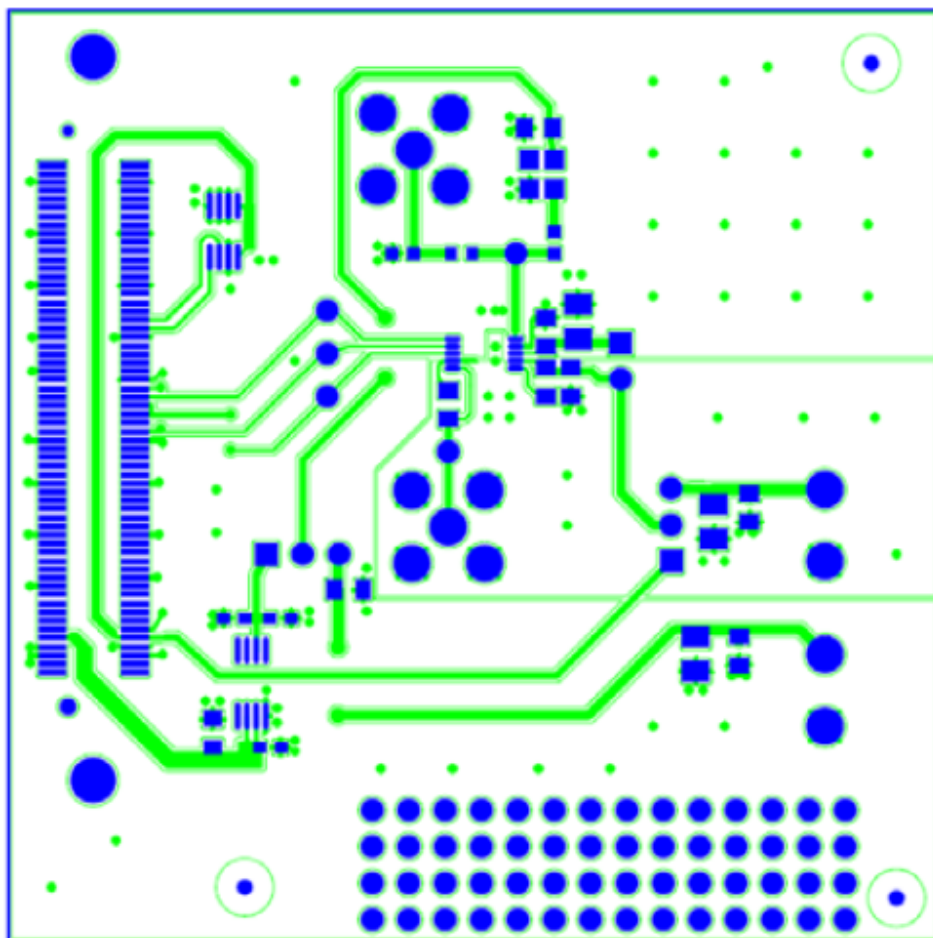


Рис. 36. AD9833 Evaluation Board, сторона компонентов.

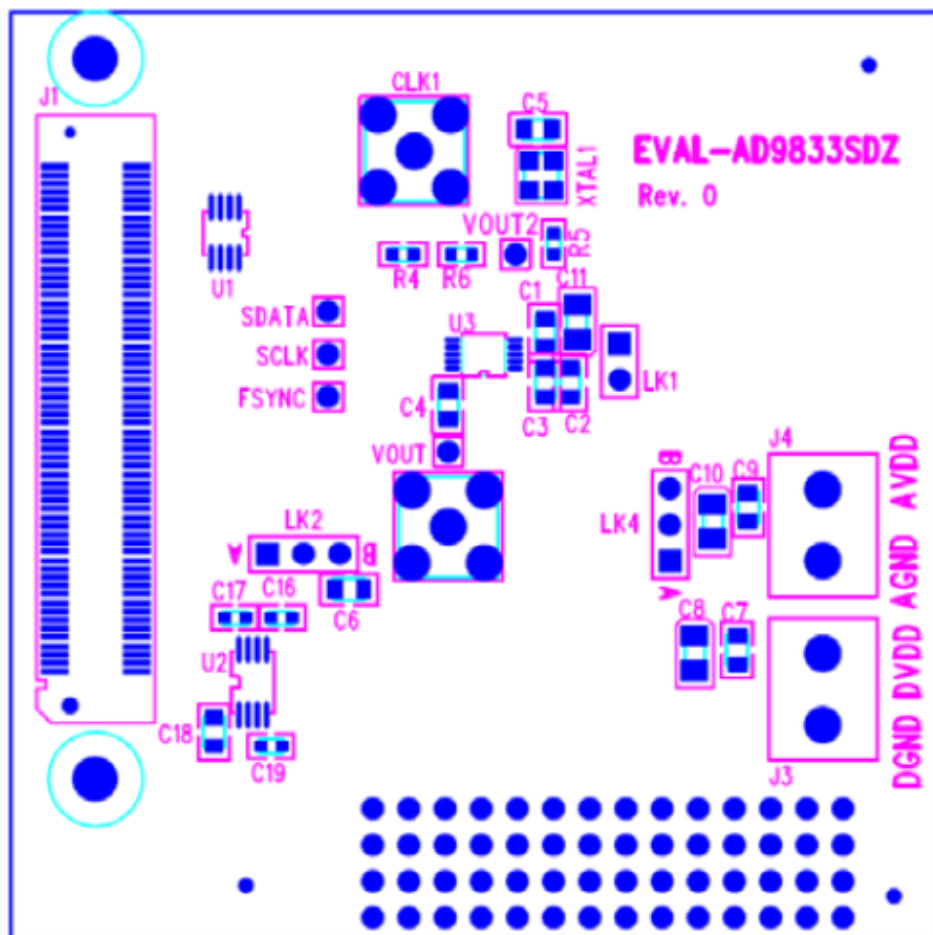


Рис. 37. AD9833 Evaluation Board, шелкография.

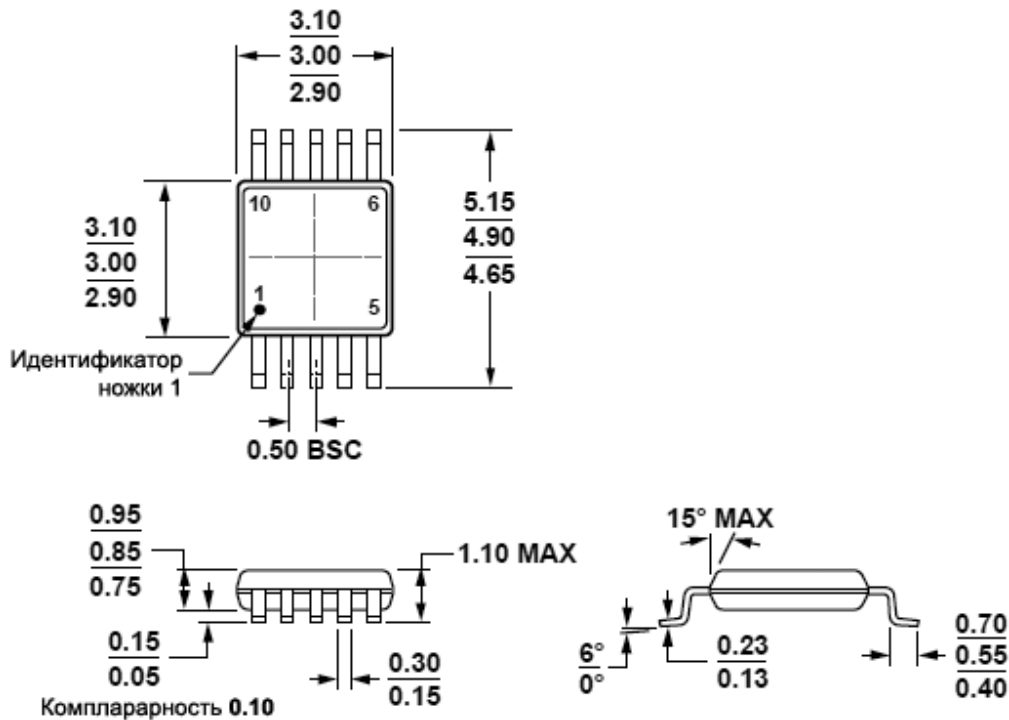


Рис. 39. 10-выводный корпус Mini Small Outline Package (MSOP, RM-10). Размеры указаны в миллиметрах (совместим со стандартами JEDEC MO-187-BA).

[Руководство по выбору при покупке]

Модель ^{1,2,3}	Диапазон температур	Корпус	Опция корпуса	Брендинг
AD9833BRM	-40°C .. +105°C	10-выводный MSOP	RM-10	DJB
AD9833BRM-REEL	-40°C .. +105°C	10-выводный MSOP	RM-10	DJB
AD9833BRM-REEL7	-40°C .. +105°C	10-выводный MSOP	RM-10	DJB
AD9833BRMZ	-40°C .. +105°C	10-выводный MSOP	RM-10	D68
AD9833BRMZ-REEL	-40°C .. +105°C	10-выводный MSOP	RM-10	D68
AD9833BRMZ-REEL7	-40°C .. +105°C	10-выводный MSOP	RM-10	D68
AD9833WBRMZ-REEL	-40°C .. +105°C	10-выводный MSOP	RM-10	D68
EVAL-AD9833SDZ		Evaluation Board		

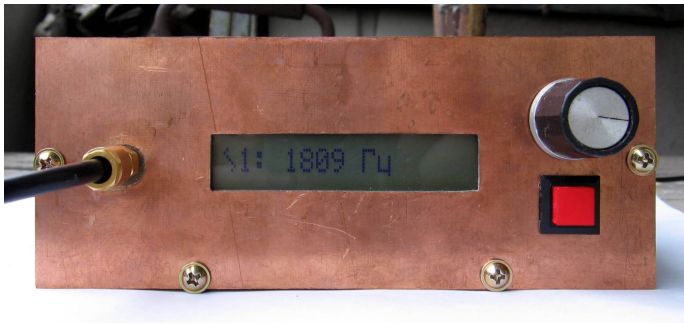
Примечания:

- 1 Z = микросхема совместима со стандартом RoHS.
- 2 W = квалифицировано для автомобильных приложений (Automotive).
- 3 Оценочная плата для AD9833 требует для запуска плату SDP, которая продается отдельно.

Вариант поставки AUTOMOTIVE. Модель AD9833WBRMZ-REEL доступна с контролем производства для поддержки качества и надежности автомобильных (automotive) приложений. Обратите внимание, что эта automotive модель может иметь спецификации, отличающиеся от коммерческих моделей; таким образом, разработчики должны тщательно заново просмотреть секцию спецификаций этого даташита. Только продукция, сертифицированная как automotive, представлена для использования в автомобильных приложениях. Свяжитесь с локальным представителем компании Analog Devices для получения информации о закупке, и получения специальных отчетов надежности микросхем для автомобильных приложений (Automotive Reliability reports).

[Самодельный генератор сигналов]

За несколько дней собрал генератор сигналов на основе платы, купленной на aliexpress, и макетной платы AVR-USB-MEGA16. Генератор имеет LCD-индикатор BC1601A1 (16x1) и простой интерфейс управления, состоящий из энкодера и кнопки.



Что умеет генератор:

- Можно менять частоту в диапазоне от 0 до 12500000 Гц с дискретностью 1 Гц.
- Можно изменять форму сигнала - синус, треугольник, прямоугольник.
- Можно регулировать амплитуду выходного сигнала. Для синусоидального и треугольного сигнала размах выходного напряжения (от пика до пика) регулируется от 0 до 3.6V с шагом 14 mV. Для прямоугольного сигнала размах выходного напряжения регулируется от 0 до 5.2V с шагом 0.124V.
- Энергонезависимая память, где хранятся настройки на 10 запрограммированных частот (вместе с формой сигнала и установленным напряжением).
- Генератор можно перепрограммировать через USB.
- Питание генератора может осуществляться через USB или от источника постоянного тока от 9 до 15V. Потребляемый ток порядка 170..180 mA.