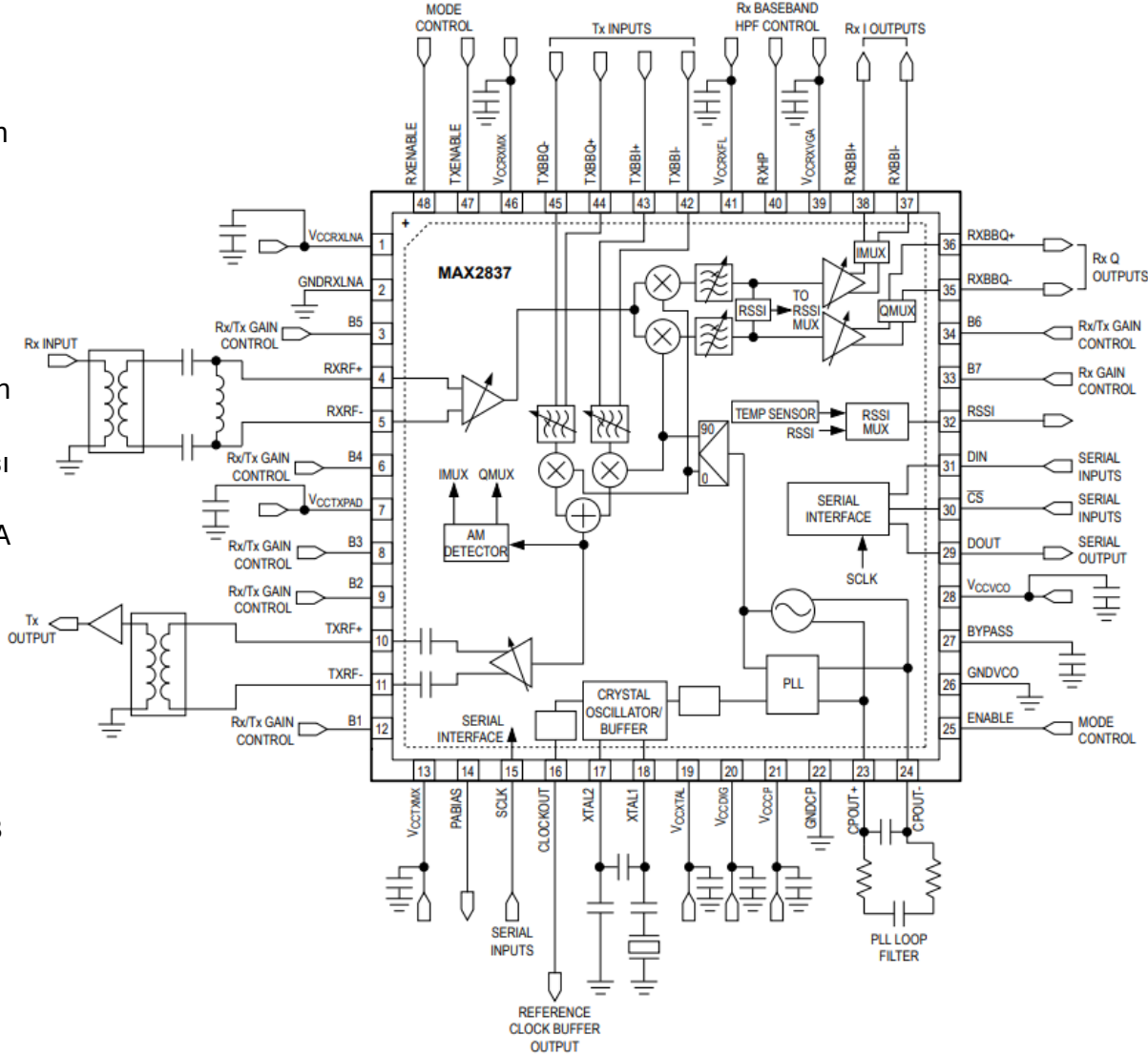


Max-2837 :

MAX2837, doğrudan dönüşüm sıfır-IF RF vericisi, özellikle 2.3GHz ile 2.7GHz arası kablosuz geniş bant sistemleri için tasarlanmıştır. MAX2837, RF vericisi işlevini uygulamak için gereken tüm devreleri tamamen entegre eder; RF'den bazbant'a alıcı yol ve bazbant'tan RF'e verici yol, VCO, frekans sentezleyici, kristal osilatör ve bazbant/kontrol arayüzü. Cihaz, daha azı 20Hz frekans adımları olan hızlı ayarlanabilen bir sigma-delta RF sentezleyiciye ve bir TCXO'nun yerine düşük maliyetli bir kristalin kullanımına olanak tanıyan bir kristal osilatörü içerir. Verici IC ayrıca, çift taraflı DC ofset iptal, I/Q hata ve taşıyıcı sızıntısı tespit devrelerini entegre eder. Yalnızca bir RF band geçiren filtre (BPF), kristal, RF anahtarı, PA ve tamamlanmış bir kablosuz geniş bant RF radyo çözümü oluşturmak için birkaç pasif bileşen gereklidir. MAX2837, alıcı ve verici için monolitik filtreler uygulayarak harici bir SAW filtresi ihtiyacını tamamen ortadan kaldırır. Bazbant filtreleri, Rx ve Tx sinyal yolları, sıkı gürültü figürü ve doğrusallık özelliklerini karşılamak için optimize edilmiştir. Cihaz, 2048 FFT OFDM'ye kadar destek sağlar ve 1.75MHz ile 28MHz RF kanal bant genişlikleri için programlanabilir kanal filtreleri uygular. Verici, frekans geçici yerleşimi de içeren yalnızca 2µs Tx-Rx geçiş süresi gerektirir. Entegre devre, yalnızca 6mm x 6mm x 0.8mm ölçülerinde küçük, 48 pinli ince QFN paketinde mevcuttur.



Features

- 2.3GHz to 2.7GHz Wideband Operation
- Complete RF Transceiver, PA Driver, and Crystal Oscillator
 - 0dBm Linear OFDM Transmit Power
 - -70dBm Tx Spectral Emission Mask
 - 2.3dB Rx Noise Figure
 - Tx/Rx I/Q Error and LO Leakage Detection
 - Monolithic Low-Noise VCO with -39dBc Integrated Phase Noise
 - Programmable Tx I/Q Lowpass Anti-Aliasing Filter
 - Sigma-Delta Fractional-N PLL with 20Hz Step Size
 - 45dB Tx Gain-Control Range
 - 94dB Receive Gain-Control Range
 - 60dB Analog RSSI Instantaneous Dynamic Range
 - 4-Wire SPI Digital Interface
 - I/Q Analog Baseband Interface
 - Digitally Tuned Crystal Oscillator
 - On-Chip Digital Temperature Sensor Read-Out
- +2.7V to +3.6V Transceiver Supply
- Low-Power Shutdown Current
- Small 48-Pin Thin QFN Package (6mm x 6mm x 0.8mm)

Max-2837 :host / libhacrf / src / hackrf.c

HACKRF_VENDOR_REQUEST_MAX2837_WRITE = 2, (63.satır) (hackrf_vendor_request isimli enum içerisinde tanımlı)

HACKRF_VENDOR_REQUEST_MAX2837_READ = 3, (64.satır)

```
typedef struct {
```

```
    uint32_t bandwidth_hz;
```

```
} max2837_ft_t;
```

```
static const max2837_ft_t max2837_ft[] = {
```

```
{1750000},{2500000},{3500000},{5000000},{5500000},{6000000},{7000000},{8000000},{9000000},{10000000},{12000000},{14000000},{15000000},{20000000},
```

```
{24000000},{28000000},{0}}; (şeklinde tanımlamada kullanılmıştır)
```

```
int ADDCALL hackrf_max2837_read(hackrf_device* device ,      uint8_t register_number ,          uint16_t* value)      (887.satır)
```

```
{
```

```
    int result;
```

```
    if (register_number >= 32) {
```

```
        return HACKRF_ERROR_INVALID_PARAM;
```

```
    }
```

```
    result = libusb_control_transfer(
```

```
        device->usb_device,
```

```
        LIBUSB_ENDPOINT_IN | LIBUSB_REQUEST_TYPE_VENDOR | LIBUSB_RECIPIENT_DEVICE,
```

```
        HACKRF_VENDOR_REQUEST_MAX2837_READ,
```

```
        0,
```

```
        register_number,
```

```
        (unsigned char*) value,
```

```
        2,
```

```
        0);
```

```
    if (result < 2) {
```

```
        last_libusb_error = result;
```

```
        return HACKRF_ERROR_LIBUSB;
```

```
    } else {
```

```
        return HACKRF_SUCCESS;
```

```
}
```

Hackrf.c -----→ 916.satır:

```
int ADDCALL hackrf_max2837_write(
    hackrf_device* device,
    uint8_t register_number,
    uint16_t value)
{
    int result;
    if (register_number >= 32) {
        return HACKRF_ERROR_INVALID_PARAM;
    }
    if (value >= 0x400) {
        return HACKRF_ERROR_INVALID_PARAM;
    }
    result = libusb_control_transfer(
        device->usb_device,
        LIBUSB_ENDPOINT_OUT | LIBUSB_REQUEST_TYPE_VENDOR | LIBUSB_RECIPIENT_DEVICE,
        HACKRF_VENDOR_REQUEST_MAX2837_WRITE,
        value,
        register_number,
        NULL,
        0,
        0);
    if (result != 0) {
        last_libusb_error = result;
        return HACKRF_ERROR_LIBUSB;
    } else {
        return HACKRF_SUCCESS;
    }
}
```

(hackrf-master\host\hackrf-tools\src\hackrf.h)

RX PATH :

MAX2837'deki bazband kazancı ("BB" veya "VGA") - 0-62dB, 2dB adımlarla, `@ref hackrf_set_vga_gain` işlevi aracılığıyla yapılandırılabilir.

MAX2837'deki RX IF kazancı ("IF") - 0-40dB, 8dB adımlarla, `@ref hackrf_set_lna_gain` işlevi aracılığıyla yapılandırılabilir

Anten bağlantı noktasına yakın RX RF yükselteci ("RF") - 0 veya ~11dB, `@ref hackrf_set_amp_enable` işleviyle etkinleştirilir veya devre dışı bırakılır (aynı işlev TX modunda TX RF amplisini etkinleştirmek/devre dışı bırakmak için kullanılır)

TX PATH :

MAX2837'deki TX IF kazancı ("IF" veya "VGA") - 0-47dB, 1dB adımlarla, `@ref hackrf_set_txvga_gain` işlevi aracılığıyla yapılandırılabilir

Anten bağlantı noktasına yakın TX RF yükselteci ("RF") - 0 veya ~11dB, `@ref hackrf_set_amp_enable` işleviyle etkinleştirilir veya devre dışı bırakılır (aynı işlev RX modunda RX RF amplisini etkinleştirmek/devre dışı bırakmak için kullanılır)

Ayarlama : HackRF One, neredeyse herhangi bir frekansa 1-6000MHz arasında (ve teorik olarak sınırlar biraz daha yüksektir) ayarlanabilir. Bu, MAX2837 transceiver IC'nin RF bölümünün RFFC5072 mikser/sentezleyicisinin yerel osilatörü ile up/down-converting yapılmasıyla elde edilir. Mikser, IF ve LO frekanslarının toplam ve fark frekanslarını üretir, ve bir LPF(low pass filter) veya HPF(high pass filter) filtresi, sonuçta oluşan frekanslardan birini seçmek için kullanılabilir. Filtreyi atlayıp IF'yi olduğu gibi kullanma olasılığı da vardır. IF ve LO frekansları bağımsız olarak programlanabilir ve davranış seçilebilir. Daha fazla ayrıntı için `@ref hackrf_set_freq_explicit` işlevine bakın.İstenen frekans için uygun LO ve IF frekanslarını ve RF yolunu otomatik olarak seçen `@ref hackrf_set_freq` adlı pratik bir işlev de vardır. Çoğu durumda bu işlev kullanılmalıdır.

Filtreleme : MAX2837'de hem RX hem de TX için dahili seçilebilir bir bazband filtresi bulunur. Genişliği `@ref hackrf_set_baseband_filter_bandwidth` ile ayarlanabilir, ancak yalnızca bazı değerler geçerlidir. Geçerli değerler, `@ref hackrf_compute_baseband_filter_bw_round_down_lt` ve `@ref hackrf_compute_baseband_filter_bw` işlevleri aracılığıyla alınabilir.

- **MAX2837 2.3 ila 2.7 GHz verici/almıcı**

Bu verici yongası HackRF One'ın RF modülatör/demodülatörüdür. Bu yonga, analog I/Q örneklerini MAX5864 ADC/DAC yongasına gönderir/alır.

Bu yongaların kayıtlarına @ref hackrf_max2837_read ve @ref hackrf_max2837_write işlevleri aracılığıyla erişilebilir.

```
extern ADDAPI int ADDCALL hackrf_set_freq(hackrf_device* device, const uint64_t freq_hz); (hackrf.h---1543.satır)
```

Bu işlev, belirli bir cihazın merkez frekansını açıkça ayarlamak için kullanılır. İşlevin parametreleri şunlardır:

device: Ayarlanacak cihaz.

if_freq_hz: MAX2837 transceiver IC'nin ayar frekansı (Hz cinsinden). Aralığı 2150 ila 2750MHz arasında olmalıdır.

lo_freq_hz: RFFC5072 mixer/sentezleyici IC'nin ayar frekansı (Hz cinsinden). Aralığı 84.375 ila 5400MHz arasında olmalıdır ve varsayılan olarak 1000MHz'dir. Eğer path @ref RF_PATH_FILTER_BYPASS olarak ayarlanmışsa bu parametre etkili olmaz.

path: Karıştırıcı için filtre yolunu belirler. Ayrıntılar için @ref rf_path_filter belgelerine bakınız.

İşlevin başarılı bir şekilde çalışması durumunda @ref HACKRF_SUCCESS değeri döndürülür. Aksi halde bir hata kodu döner.

extern ADDAPI int ADDCALL hackrf_board_partid_serialno_read(hackrf_device* device, read_partid_serialno_t* read_partid_serialno); (hackrf.h----1621)

Bu işlem, belirli bir cihazın RF alıcı kazançını (MAX2837 transceiver IC'nin "IF" kazanç ayarı) belirler. İşlevin parametreleri şunlardır:

device: Yapılandırılacak cihaz.

value: dB cinsinden RX IF kazanç değeri. Aralığı 0 ila 40 dB arasında olmalıdır ve 8 dB adımlarıyla ayarlanabilir.

İşlev, başarıyla tamamlandığında @ref HACKRF_SUCCESS değerini döndürür. Aksi takdirde bir hata kodu döner.

extern ADDAPI int ADDCALL hackrf_set_lna_gain(hackrf_device* device, uint32_t value); (hackrf.h----1635)

Bu işlem, belirli bir cihazın MAX2837 transceiver IC'nin ("BB" veya "VGA" kazanç ayarı) bazband alıcı kazancını ayarlar. İşlevin parametreleri şunlardır:

- `device`: Yapılandırılacak cihaz.

- `value`: dB cinsinden RX BB kazanç değeri. Aralığı 0 ila 62 dB arasında olmalıdır ve 2 dB adımlarıyla ayarlanabilir.

İşlev, başarıyla tamamlandığında @ref HACKRF_SUCCESS değerini döndürür. Aksi takdirde, bir hata kodu döner.

extern ADDAPI int ADDCALL hackrf_set_vga_gain(hackrf_device* device, uint32_t value);

Bu işlem, belirli bir cihazın MAX2837 transceiver IC'nin ("IF" veya "VGA" kazanç ayarı) RF verici kazancını ayarlar. İşlevin parametreleri şunlardır:

- `device`: Yapılandırılacak cihaz.

- `value`: dB cinsinden TX IF kazanç değeri. Aralığı 0 ila 47 dB arasında olmalıdır ve 1 dB adımlarıyla ayarlanabilir.

İşlev, başarıyla tamamlandığında @ref HACKRF_SUCCESS değerini döndürür. Aksi takdirde, bir hata kodu döner.

Si5351

Si5351, I2C ile yapılandırılabilen bir saat üreticidir ve kristalleri, kristal osilatörleri, VCXO'ları, faz kilitleme döngülerini (PLL'ler) ve fanout tamponlarını maliyet duyarlı uygulamalarda ideal bir şekilde değiştirmek için uygundur. PLL/VCXO + yüksek çözünürlüklü MultiSynth kesirli bölen mimarisi temel alınarak, Si5351 her bir çıkışında 0 ppm hatayla 200 MHz'ye kadar herhangi bir frekans üretebilir. Si5351'in çeşitli uygulamaları karşılamak üzere üç versiyonu bulunmaktadır. Si5351A, kristalleri ve kristal osilatörlerini değiştirmek için dahili bir osilatör kullanarak 8 serbest çalışan saat üretebilir. Si5351B, dahili bir VCXO ekler ve hem serbest çalışan saatleri hem de senkron saatleri değiştirme esnekliği sağlar. Özel çekilebilir kristaller için daha yüksek maliyetli olmasını engellerken geniş bir ayarlama aralığında güvenilir çalışma sağlar. Si5351C, aynı esnekliği sunar ancak harici bir referans saatine (CLKIN) senkronize olur.

-Giriş aşaması, cihazın sürümüne (A/B/C) bağlı olarak harici bir kristal (XTAL), bir kontrol gerilim girişi (VC) veya bir saat girişi (CLKIN) kabul eder. Sentez aşamasının ilk aşaması, giriş frekanslarını yüksek frekanslı bir ara saate çarpar, sentezin ikinci aşaması ise istenen çıkış frekanslarını oluşturmak için yüksek çözünürlüklü MultiSynth kesirli bölenleri kullanır. Çıkış aşamasında, çıkış frekanslarını 2.5 kHz gibi düşük değerlere kadar oluşturmak için ek tam bölmeler sağlanmaktadır. Sentez aşamalarındaki her bir çapraz anahtar, girişlerin çıktılara yönlendirilmesinde tam esneklik sağlar.

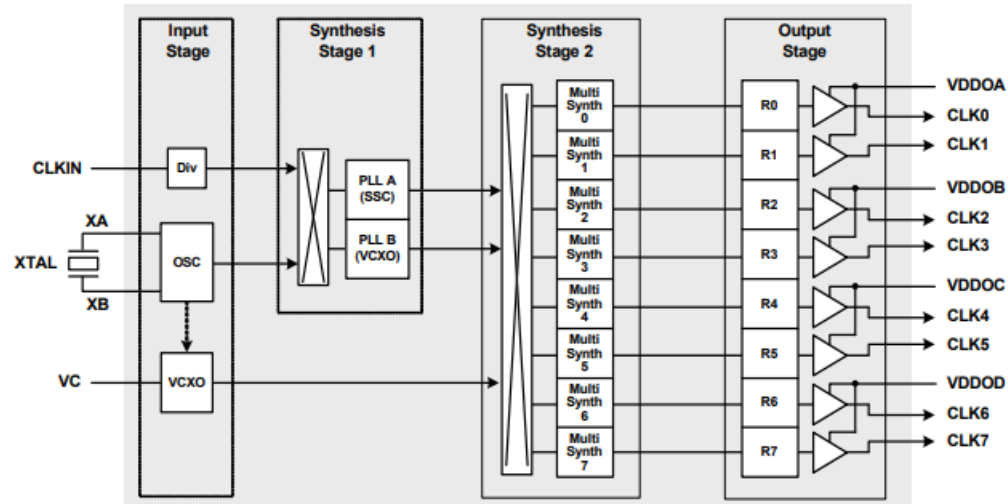


Figure 3. Si5351 Block Diagram

Si5351, tamamen yapılandırılabilir bir şekilde çalışan son derece esnek bir saat üreticidir ve bu özelliklerinin tümü I2C arabirimi aracılığıyla yapılandırılabilir. Cihazın varsayılan yapılandırması, kalıcı bellekte (NVM) saklanır. NVM, bir kez yazılabilir bellek (OTP) olarak adlandırılan bir bellektir ve özel bir kullanıcı yapılandırmasını güç verildiğinde saklayabilir. Bu, güç verildiğinde bir saat gerektiren uygulamalar için (örneğin, bir işlemciye bir saat sağlamak için) kullanışlı bir özelliktir.

Hackrf.c dosyasındaki si5351 kodları ve açıklamaları :

HACKRF_VENDOR_REQUEST_SI5351C_WRITE = 4, (65)

HACKRF_VENDOR_REQUEST_SI5351C_READ = 5, (66)

int ADDCALL hackrf_si5351c_read(hackrf_device* device, uint16_t register_number, uint16_t* value) (si5351 deki verileri okuma kodu (949))

{

uint8_t temp_value;

int result;

if (register_number >= 256) {

return HACKRF_ERROR_INVALID_PARAM;

}

temp_value = 0;

result = libusb_control_transfer (device->usb_device, LIBUSB_ENDPOINT_IN | LIBUSB_REQUEST_TYPE_VENDOR | LIBUSB_RECIPIENT_DEVICE,
HACKRF_VENDOR_REQUEST_SI5351C_READ, 0, register_number, (unsigned char*) &temp_value, 1, 0);

if (result < 1) {

last_libusb_error = result;

return HACKRF_ERROR_LIBUSB;

} else {

*value = temp_value;

return HACKRF_SUCCESS;

}

```
int ADDCALL hackrf_si5351c_write( hackrf_device* device ,  uint16_t register_number ,  uint16_t value)      (981)
{
    int result;
    if (register_number >= 256) {
        return HACKRF_ERROR_INVALID_PARAM;
    }
    if (value >= 256) {
        return HACKRF_ERROR_INVALID_PARAM;
    }
    result = libusb_control_transfer( device->usb_device , LIBUSB_ENDPOINT_OUT | LIBUSB_REQUEST_TYPE_VENDOR | LIBUSB_RECIPIENT_DEVICE
    HACKRF_VENDOR_REQUEST_SI5351C_WRITE , value ,    register_number , NULL ,    0 , 0);
    if (result != 0) {
        last_libusb_error = result;
        return HACKRF_ERROR_LIBUSB;
    } else {
        return HACKRF_SUCCESS;
    }
}
```

Hackrf.h dosyasındaki si5351 kodları ve açıklamaları :

Si5351C Saat Üretici

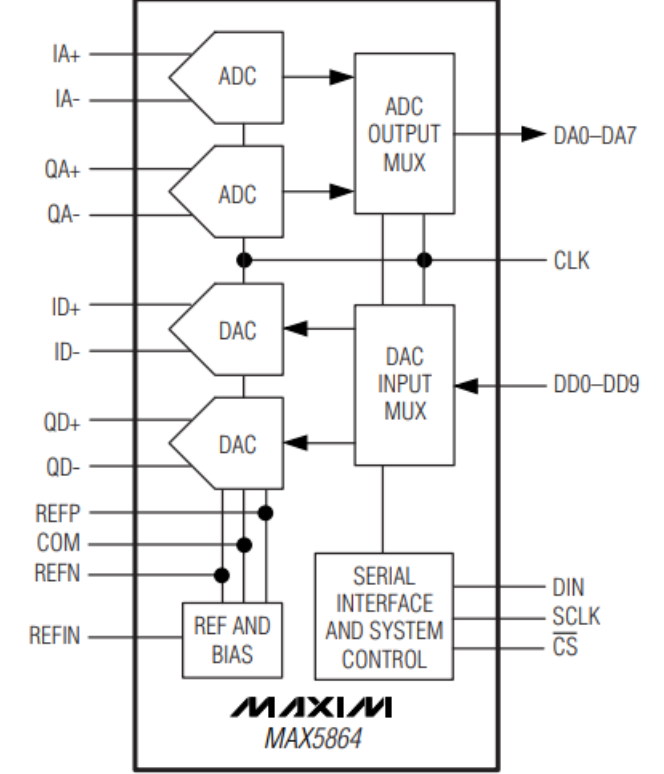
* Bu yonga, diğer tüm yongalara saat sinyalleri sağlar. Saat girişlerinden (dahili veya harici) geniş bir frekans aralığı sentezleyebilir. Bir sabit 800 MHz dahili saat kullanır (bir PLL aracılığıyla sentezlenir).

* Kayıtlarına, [@ref hackrf_si5351c_read](#) ve [@ref hackrf_si5351c_write](#) işlevleri aracılığıyla erişilebilir.

MAX5864 ADC/DAC

- Bu yonga, alınan analog I-Q örneklerini dijitale ve iletilen I-Q örneklerini analoge dönüştürür. Ana ARM MCU'ya CPLD aracılığıyla bağlanır. Yapılandırma gerektirmez, sadece örnekleme hızı saat üretici IC aracılığıyla ayarlanabilir .

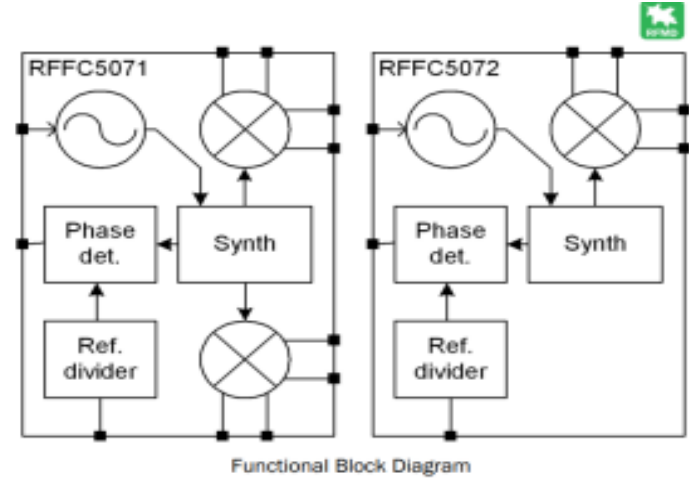
MAX5864, ultra düşük güç tüketimine sahip, yüksek entegreli bir analog ön uçtur. Taşınabilir iletişim ekipmanları için idealdir ve el cihazları, PDAlar, WLAN ve 3G kablosuz terminaller gibi cihazlarda kullanılır. **MAX5864**, çift 8-bit alıcı ADC ve çift 10-bit verici DAC'ları entegre ederken ultra düşük güçte en yüksek dinamik performansı sağlar. ADC'lerin analog I-Q giriş amplifikatörleri tamamen diferansiyel olup 1VP-P tam ölçekli sinyalleri kabul eder. Tipik I-Q kanal faz eşleşmesi $\pm 0.1^\circ$ ve amplitüt eşleşmesi $\pm 0.03\text{dB}$ 'dir. ADC'ler, $f_{IN} = 5.5\text{MHz}$ ve $f_{CLK} = 22\text{Msps}$ 'de 48.5dB SINAD ve 69dBc spuri olmayan dinamik aralık (SFDR) sunar. DAC'ların analog I-Q çıkışları tamamen diferansiyeldir ve $\pm 400\text{mV}$ tam ölçekli çıkışa sahiptir, ayrıca 1.4V ortak mod seviyesine sahiptir. Tipik I-Q kanal faz eşleşmesi $\pm 0.15^\circ$ ve amplitüt eşleşmesi $\pm 0.05\text{dB}$ 'dir. DAC'lar ayrıca 71.7dBc SFDR ile çift 10-bit çözünürlüğe sahiptir ve $f_{OUT} = 2.2\text{MHz}$ ve $f_{CLK} = 22\text{MHz}$ 'de 57dB SNR sunar. ADC'ler ve DAC'lar frekans bölme çift yönlü (FDD) ve zaman bölme çift yönlü (TDD) modlarında eşzamanlı veya bağımsız olarak çalışır. 3 tel seri arayüzü, güç kapatma ve verici işlem modlarını kontrol eder. Tipik çalışma gücü, ADC'lerin ve DAC'ların transceiver modunda eşzamanlı olarak çalıştığı $f_{CLK} = 22\text{Msps}$ 'de 42mW'dir. MAX5864, tüm çalışma güç kaynağı aralığında ve sıcaklık aralığında kararlı olan dahili 1.024V gerilim referansına sahiptir. MAX5864, +2.7V ila +3.3V aralığında bir analog güç kaynağında ve mantıksal uyumluluk için +1.8V ila +3.3V aralığında bir dijital I/O güç kaynağında çalışır. Boşta akım 5.6mA ve kapatma modunda 1 μA 'dır.



Max5864 entegresinin hackrf.h ve hackrf.c dosyalarında kodu bulunmamaktadır

RFFC5072A, birçok farklı işlevi tek bir entegre devre içinde birleştirir. Ana özellikleri arasında RF frekanslarını alarak, işleyerek ve dijital sinyallere dönüştürerek bu sinyalleri daha düşük frekanslara dönüştürmek bulunur. Bu çip, özellikle geniş bir frekans aralığında çalışabilir ve yüksek hassasiyet ve hız sağlar.

-yüksek frekanslı RF sinyallerini işlemek ve daha düşük frekanslarda dijital işlemciler veya başka bir işlemci ile işlenmesi için uygun bir formata dönüştürmek için kullanılır.



85 MHz ila 4200 MHz LO Frekans Aralığı

Tipik Adım Boyutu 1.5 Hz

Tamamen Entegre Düşük Faz Gürültülü VCO ve LO buffer (tampon bölge)

Yüksek Doğrusallıkta RF Karıştırıcı(lar) (mixer)

30 MHz ila 6000 MHz Karıştırıcı Frekans Aralığı

Giriş IP3 +23 dBm

Karıştırıcı Bias Ayarlanabilir Düşük Güç İşlemi için

Tam Dupleks Modu (RFFC5071)

2.7 V ila 3.3 V Güç Kaynağı

Düşük Akım Tüketimi

Features

- 85MHz to 4200MHz LO Frequency Range
- Fractional-N Synthesizer with Very Low Spurious Levels
- Typical Step Size 1.5Hz
- Fully Integrated Low Phase Noise VCO and LO Buffers
- High Linearity RF Mixer(s)
- 30MHz to 6000MHz Mixer Frequency Range
- Input IP3 +23dBm
- Mixer Bias Adjustable for Low Power Operation
- Full Duplex Mode (RFFC5071)
- 2.7V to 3.3V Power Supply
- Low Current Consumption
- 3- or 4-Wire Serial Interface

RFFC5072 nin hackrf.c dosyasında kullanıldığı yerler ve açıklamalar

HACKRF_VENDOR_REQUEST_RFFC5071_WRITE = 8, (hackrf.c --- 69.satır)

HACKRF_VENDOR_REQUEST_RFFC5071_READ = 9, (hackrf.c --- 70.satır)

```
-----  
  
int ADDCALL hackrf_rffc5071_read(    hackrf_device* device,    uint8_t register_number,    uint16_t* value)    (hackrf.c --- 1038.satır)  
{  
  
    int result;  
  
    if (register_number >= 31) {  
  
        return HACKRF_ERROR_INVALID_PARAM;    }  
  
    result = libusb_control_transfer(    device->usb_device,    LIBUSB_ENDPOINT_IN | LIBUSB_REQUEST_TYPE_VENDOR | LIBUSB_RECIPIENT_DEVICE,  
        HACKRF_VENDOR_REQUEST_RFFC5071_READ,    0,    register_number,    (unsigned char*) value,    2,    0);  
  
    if (result < 2) {  
  
        last_libusb_error = result;  
  
        return HACKRF_ERROR_LIBUSB;  
  
    } else {  
  
        return HACKRF_SUCCESS;  
  
    }  
  
}
```

Hackrf üzerindeki rffc5072 'nin değerlerini okumasını sağlayan kod

```
int ADDCALL hackrf_rffc5071_write(    hackrf_device* device,    uint8_t register_number,    uint16_t value)
{
    int result;
    if (register_number >= 31) {
        return HACKRF_ERROR_INVALID_PARAM;
    }
    result = libusb_control_transfer(    device->usb_device,    LIBUSB_ENDPOINT_OUT |    LIBUSB_REQUEST_TYPE_VENDOR |
        LIBUSB_RECIPIENT_DEVICE,    HACKRF_VENDOR_REQUEST_RFFC5071_WRITE,
        value,    register_number,    NULL,    0,    0);
    if (result != 0) {
        last_libusb_error = result;
        return HACKRF_ERROR_LIBUSB;
    } else {
        return HACKRF_SUCCESS;
    }
}
```

Hackrf üzerindeki rffc5072 'nin değerlerini okuyup host koduna alan fonksiyon

RFFC5072 nin hackrf.c dosyasında kullanıldığı yerler ve açıklamalar

RFFC5072 Sentezleyici/mikser

* Bu mikser, RF sinyalinin dahili olarak sentezlenmiş yerel osilatör sinyali ile karıştırır ve bu şekilde toplam ve fark frekanslarını elde eder. LPF veya HPF filtreleri ile birleştirilmiş ve MAX2837 IC'deki frekans ayarı ile birlikte herhangi bir frekansa ayarlanabilir.

* Kayıtları, @ref hackrf_rffc5071_read ve @ref hackrf_rffc5071_write işlevleri aracılığıyla erişilebilir.

extern ADDAPI int ADDCALL hackrf_rffc5071_read(hackrf_device* device, uint8_t register_number, uint16_t* value); (1371. satırdaki kod)

* RFFC5071/5072 mikser-sentezleyici IC'nin kayıtlarını doğrudan okur

Sadece hata ayıklama amaçları için tasarlanmıştır!

@param[in] device sorgulanacak cihaz

@param[in] register_number okunacak kayıt numarası

@param[out] value belirtilen kaydın değeri

@return başarı durumunda @ref HACKRF_SUCCESS veya @ref hackrf_error varyantı

@ingroup debug

extern ADDAPI int ADDCALL hackrf_rffc5071_write(hackrf_device* device, uint8_t register_number, uint16_t value);

* RFFC5071/5072 mikser-sentezleyici IC'nin kayıtlarına doğrudan yazma işlemi yapar

Sadece hata ayıklama amaçları için tasarlanmıştır!

@param[in] device yazılacak cihaz

@param[in] register_number yazılacak kayıt numarası

@param[out] value belirtilen kayda yazılacak değer

@return başarı durumunda @ref HACKRF_SUCCESS veya @ref hackrf_error varyantı

@ingroup debug

Hackrf.h dosyasındaki -----1545.satırdaki kodlar

```
extern ADDAPI int ADDCALL hackrf_set_freq_explicit( hackrf_device* device,  const uint64_t if_freq_hz,  
                                                    const uint64_t lo_freq_hz,          const enum rf_path_filter path);
```

Belirli bir ayarla merkez frekansını ayarlar

Merkez frekansı, $f_{center} = f_{IF} + k \cdot f_{LO}$ olacak şekilde ayarlanır, burada $k \in \{-1; 0; 1\}$, @p path değerine bağlıdır. Ayrıntılar için @ref rf_path_filter belgelerine bakın.

@param device ayarlanacak cihaz

@param if_freq_hz MAX2837 verici IC'nin ayarlama frekansı Hz cinsinden. 2150-2750MHz aralığında olmalıdır

@param lo_freq_hz RFFC5072 mikser/sentezleyici IC'nin ayarlama frekansı Hz cinsinden. 84.375-5400MHz aralığında olmalıdır, varsayılan 1000MHz'dir. @p path @ref RF_PATH_FILTER_BYPASS olarak ayarlanmışsa etkisi yoktur

@param path mikser için filtre yolunu belirler. Ayrıntılar için @ref rf_path_filter belgelerine bakın

@return başarı durumunda @ref HACKRF_SUCCESS veya @ref hackrf_error varyantı

@ingroup configuration

XC2C64A – cpld ‘nin hackrf.h içerisindeki kodların ve açıklamaları

* ## XC2C64A CPLD

- Bu CPLD, MAX5864 ADC/DAC ve ana MCU arasında yer alır ve genellikle veri formatı dönüşümü ve bazı senkronizasyon işlemleri gerçekleştirir.

extern ADDAPI int ADDCALL hackrf_cpld_write(hackrf_device* device, unsigned char* const data, const unsigned int total_length); (462.satırdaki kod)

hackrf_cpld_write işleminden sonra cihaz sıfırlanmalıdır

XC2C64A-7VQ100C CPLD içine yapılandırma bit akışını yazar

@deprecated bu işlev bit akışını yazar, ancak firmware her sıfırlamada otomatik olarak geçersiz kıldığından dolayı herhangi bir değişiklik etkili olmayacaktır

@param device yapılandırılacak cihaz

@param data CPLD bit akışı verisi

@param total_length yazılacak bit akışının uzunluğu

@return başarı durumunda @ref HACKRF_SUCCESS veya @ref hackrf_error varyantı

@ingroup debug

Hackrf.c dosyasındaki cpld kodları ve açıklamaları :

HACKRF_VENDOR_REQUEST_CPLD_CHECKSUM = 36, (95.satırdaki kod)

TRANSCIVER_MODE_CPLD_UPDATE = 4, (120.satırdaki kod)

```
int ADDCALL hackrf_cpld_write(    hackrf_device* device,          unsigned char* const data, const unsigned int total_length)
{
    const unsigned int chunk_size = 512;

    unsigned int i;

    int result, transferred = 0;

    result = hackrf_set_transceiver_mode(device, TRANSCIVER_MODE_CPLD_UPDATE);

    if (result != 0)

        return result;

    for (i = 0; i < total_length; i += chunk_size) {
result = libusb_bulk_transfer(device->usb_device,    TX_ENDPOINT_ADDRESS,    &data[i],    chunk_size,    sferred,    10000 );

        if (result != LIBUSB_SUCCESS) {

            last_libusb_error = result;

            return HACKRF_ERROR_LIBUSB;

        }

    }

    return HACKRF_SUCCESS;

}
```