

2.4 GHZ IEEE 802.15.4/ZIGBEE® RF トランシーバ

アプリケーション

- IEEE 802.15.4 システム
- ZigBee® システム
- 産業用の監視および制御機能
- ホーム・オートメーションおよびビルのオートメーション
- 自動メーター読み取り器
- ローパワー・ワイヤレス・センサ・ネットワーク
- セットトップ・ボックスおよびリモート・コントローラ
- コンシューマ・エレクトロニクス

主要な機能

- 最新の選択度および共存性
隣接チャンネル除去: 49dB
相互チャンネル除去: 54dB
- 優れたリンク・バジェット(103dB)
400m の通信範囲
- 拡張された動作温度範囲(-40°C ~ +125°C)
- 広い供給電圧範囲: 1.8V ~ 3.8V
- 広範な IEEE 802.15.4 MAC ハードウェア・サポートによりマイクロ・コントローラの負荷を軽減
- AES-128 セキュリティ・モジュール
- CC2420 インターフェイス互換性モード

低消費電力

- RX(フレーム受信、-50dBm) 18.5mA
- +5dBm で TX 33.6mA
- 0dBm で TX 25.8mA
- パワー・ダウンでは 1 μA 未満

全般

- 水晶 1 個で全てシステムのクロックを出力
- RoHS 準拠 5 x 5mm QFN28(RHD) パッケージ

無線

- データ・レート、250kbps の IEEE 802.15.4 準拠 DSSS ベースバンド・モデム
- 高感度なレシーバ(-98dBm)
- 最大 +5dBm のプログラマブル出力電力
- RF 周波数範囲 23947MHz ~ 2507MHz
- 世界の無線周波数割り当てに準拠を目的とするシステム向き: ETSI EN 300 328 および EN 300 440 クラス 2(ヨーロッパ)、FCC CFR47 パート 15(米国)、および ARIB STD-T66 (日本)

マイクロコントローラ・サポート

- デジタル RSSI/LQI サポート
- CSMA/CA の自動クリア・チャンネル・アセスメント
- 自動 CRC
- フレキシブル・バッファリングおよびセキュリティ・プロセッシング用の 768 バイトの RAM
- MAC セキュリティの完全サポート
- 4 線式 SPI
- 6 本の構成可能な IO ピン
- 割り込みジェネレータ
- フレーム・フィルタリングおよびプロセッシング・エンジン
- 乱数ジェネレータ

開発ツール

- リファレンス・デザイン
- IEEE 802.15.4 MAC ソフトウェア
- ZigBee® スタック・ソフトウェア
- 完全装備の開発キット
- ハードウェアで Packet Sniffer をサポート

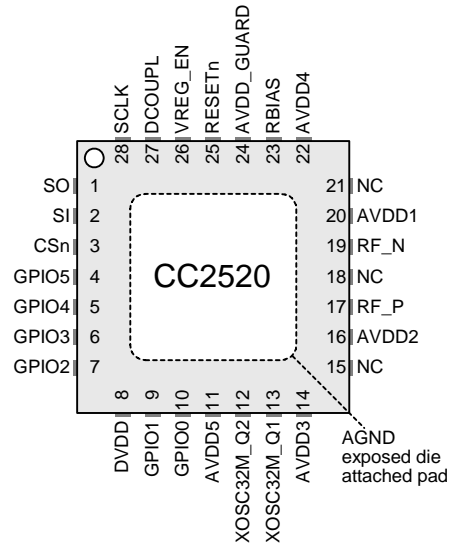
概要

CC2520 は TI の第 2 世代 ZigBee® /2.4GHz ISM 帯 IEEE 802.15.4 RF トランシーバです。このチップは、最新の選択度、優れたリンク・バジェット、最高 125°C での動作、および低電圧での動作を兼ね備えて産業用温度範囲のアプリケーションを実現します。

さらに、CC2520 では、フレーム処理、データ・バッファリング、パースト送信、データの暗号化、データ認証、クリア・チャンネル・アセスメント、リンク品質情報、およびフレーム・タイミング情報用に広範なハードウェア・サポートを提供しています。これらの機能により、ホスト・コントローラの負荷が軽減されます。

一般的なシステムの場合、CC2520 は、マイクロコントローラおよびいくつかの追加パッシブ・コンポーネントと併用されます。

QFN28(RHD)パッケージ
上面図



1 略語

AAF	アンチ・エイリアシング・フィルタ
ACK	アクノリッジ
ADC	AD コンバータ
ADI	AD インターフェイス
AES	次世代標準暗号化方式
AGC	自動ゲイン制御
AM	アクティブ・モード
ARIB	Association of Radio Industries and Businesses(電波産業会)
BER	ビット・エラー・レート
BIST	セルフ・テスト内蔵
CBC-MAC	暗号ブロック・チェイニング・メッセージ認証コード
CCA	クリア・チャンネル・アセスメント
CCM	カウンタ・モード + CBC-MAC
CDM	デバイス帯電モデル
CFR	連邦規則集
CHP	チャージ・ポンプ
CMOS	相補型金属酸化膜半導体
CRC	サイクリック冗長性検査
CSMA-CA	キャリア・センス多重アクセス/衝突検出方式
CTR	カウンタ・モード(暗号化)
CW	無変調連続波
DAC	DA コンバータ
DC	直流
DPU	データ処理装置
DSSS	直接拡散方式
ECB	電子コード・ブック(AES 動作のモード)
ESD	静電気放電
ESR	等価直列抵抗
ETSI	European Telecommunications Standards Institute(欧州電気通信標準化機構)
EU	欧州連合
EVM	エラー・ベクトル振幅
FCC	Federal Communications Commission(連邦通信委員会)
FCF	フレーム制御フィールド
FCS	フレーム・チェック・シーケンス
FFCTRL	FIFO およびフレーム制御
FIFO	先入れ先出し
FS	周波数シンセサイザ
FSM	有限状態機械
GPIO	汎用 IO
HBM	人体モデル
HSSD	高速シリアル・デバッグ
I/O	入出力
I/Q	同位相/直角位相
IEEE	Institute of Electrical and Electronics Engineers(米国電気電子学会)
IF	中間周波数
ISM	産業、科学、および医療
ITU-T	International Telecommunication Union(国際電気通信連合) – Telecommunication Standardization Sector(電気通信標準化部門)

kbps	キロビット/秒
LB	ループ・バック
LF	ループ・フィルタ
LNA	低ノイズ・アンプ
LO	局部発振器
LPF	ローパス・フィルタ
LPM	低消費電力モード
LQI	リンク品質情報
LSB	最下位ビット/バイト
LUT	参照テーブル
MAC	メディア・アクセス・コントローラ
MCU	マイクロ・コントローラ・ユニット
MFR	MAC フッタ
MHR	MAC ヘッダ
MIC	メッセージ完全性コード
MISO	マスタ・イン・スレーブ・アウト
MM	マシン・モデル
MOSI	マスタ・アウト・スレーブ・イン
MPDU	MAC プロトコル・データ・ユニット
MSB	最上位ビット/バイト
MSDU	MAC サービス・データ・ユニット
NA	利用不可
NC	未接続
O-QPSK	オフセット - 四位相偏移変調
PA	パワー・アンプ
PAN	パーソナル・エリア・ネットワーク
PCB	プリント回路基板
PD	パワー・ダウン、フェーズ・デテクタ
PER	パケット・エラー・レート
PHR	PHY ヘッダ
PHY	物理層
PLL	フェーズ・ロック・ループ
PQFP	プラスチック・クワッド・フラット・パック
PSDU	PHY サービス・データ・ユニット
PUE	プルアップ・イネーブル
QLP	クワッド・リードレス・パッケージ
RAM	ランダム・アクセス・メモリ
RBW	分解能帯域幅
RF	無線周波数
RHD	略語ではありません。これは TI で使用されるパッケージ名です。
RISC	縮小命令セット・コンピュータ
RoHS	特定有害物質の使用制限指令
ROM	読み取り専用メモリ
RSSI	受信信号強度
RX	受信
SFD	フレーム開始部
SHR	同期ヘッダ
SI	シリアル入力
SO	シリアル出力
SPI	シリアル・ペリフェラル・インターフェイス
S-PQFP	プラスチック・クワッド・フラット・パック
T/R	送信/受信

TBD	未定/未定義
TX	送信
UI	ユーザ・インターフェイス
VCO	電圧制御発振器
VGA	可変ゲイン・アンプ
XOSC	クリスタル・オシレータ
LR	低レート
NaN	数字以外

2 参照

[1] IEEE std. 802.15.4 - 2003: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications for Low Rate Wireless Personal Area Networks (LR-WPANs)
<http://standards.ieee.org/getieee802/download/802.15.4-2003.pdf>

[2] IEEE std. 802.15.4 - 2006: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications for Low Rate Wireless Personal Area Networks (LR-WPANs)
<http://standards.ieee.org/getieee802/download/802.15.4-2006.pdf>

[3] CC2420 datasheet
<http://www.ti.com/lit/pdf/swrs041>

[4] NIST FIPS Pub 197: Advanced Encryption Standard (AES), Federal Information Processing Standards Publication 197, US Department of Commerce/N.I.S.T., November 26, 2001.
<http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf>

[5] CC2520 reference designs
<http://focus.ti.com/docs/prod/folders/print/cc2520.html#applicationnotes>

[6] CC2520 Errata note
<http://www.ti.com/lit/pdf/swrz024>

[7] CC2520 Product folder
<http://focus.ti.com/docs/prod/folders/print/cc2520.html>

[8] NIST software package for randomness testing:
<http://csrc.nist.gov/rng/>

[9] The diehard software package for randomness testing:
<http://stat.fsu.edu/~geo/diehard.html>

[10] MSP430F2618 Product folder
<http://focus.ti.com/docs/prod/folders/print/msp430f2618.html>

[11] 2.4 GHz Inverted F Antenna
<http://www.ti.com/lit/pdf/swru120>

[12] Antenna selection guide
<http://www.ti.com/lit/pdf/swra161>

3 特長

2394-2507MHz トランシーバ

- DSSS トランシーバ
- 250kbps データ・レート、2M チップ・チップ・レート
- ハーフ・サイン・パルス・シェーピングを用いた O-QPSK 変調
- 超低消費電流
 - RX(フレーム受信、-50dBm): 18.5mA
 - RX(フレーム待機): 22.3mA
 - TX(+5dBm 出力電力): 33.6mA
 - TX(0dBm 出力電力): 25.8mA
- 消費電力を削減するための柔軟性の高い 3 つの電力モード
- 低消費電力の完全スタティック動作 CMOS 設計
- 最高の感度(-98dBm)
- 強力な隣接チャンネル除去(49dB)
- 強力な相互チャンネル除去(54dB)
- オンチップ VCO、LNA、PA およびフィルタ
- 低電源電圧(1.8V ~ 3.8V)
- 最大 +5dBm のプログラマブル出力電力
- I/Q ダイレクト・コンバージョン・トランシーバ

小型

- QFN 28(RHD)パッケージ、5mm x 5mm
- きわめて少ない外部コンポーネント
 - 最低数のパッシブ部品
 - リファレンス・クリスタルのみ必要
- システムのクリスタル数が限定できる、他の IC 用クロック出力
- 外部フィルタ不要

使いやすく柔軟性の高いユーザ・インターフェイス

- 4 線式 SPI
- 最大 8MHz のシリアル・クロック
- 柔軟性の高い 6 個の GPIO ピン
- 割り込みジェネレータ
- 複数イベントに対する自動応答の完全制御
- 組み込み Packet Sniffer モード
- CC2420 互換性モード

高度なデータ処理向けデータ処理装置

- 容量の大きい(768 バイト)オンチップ RAM により、強力なオンチップ・フレーム処理が可能
- 128 バイト送信データ FIFO
- 128 バイト受信データ FIFO
- RAM に対する完全な読み取りおよび書き込みアクセス権
- 128 ビット AES

IEEE 802.15.4 MAC ハードウェア・サポート

- 自動プリアンブル・ジェネレータ
- 同期ワードの挿入と削除
- MAC ペイロードでの CRC-16 計算と検証
- フレーム・フィルタリング
- 自動 ACK と保留ビットの設定

- クリア・チャンネル・アセスメント(CCA)
- エネルギー検出/RSSI
- リンク品質情報(LQI)
- 完全自動 MAC セキュリティ(CTR、CBC-MAC、CCM)

開発ツール

- プロダクト・フォルダ[7]を参照

以下の基準に準拠を目的とするシステム向け

- IEEE 802.15.4 PHY
- ETSI EN 300 328
- ETSI EN 300 440 クラス 2
- FCC CFR47 パート 15
- ARIB STD-T66

4 絶対最大定格

動作(自由通気)温度範囲内にて(特に記述のない限り)⁽¹⁾

パラメータ	制限	単位
電源電圧 ⁽²⁾	-0.3 ~ 3.9	V
デジタル・ピンの電圧	-0.3 ~ VDD + 0.3(最大 3.9)	V
1.8V ピンの電圧	-0.3 ~ 2.0	V
入力 RF レベル	+10	dBm
保存温度範囲	-50 ~ 150	°C
リフロー半田付け温度	260	°C
ESD HBM	800	V
ESD CDM	500	V
ESD MM	100	V

- 1) 「絶対最大定格」以上のストレスは、製品に恒久的なダメージを製品に与えることがあります。これはストレスの定格のみについて示してあり、このデータシートの「推奨動作条件」に示された値を越える状態での本製品の機能動作を意味するものではありません。絶対最大定格の状態に長時間置くと、本製品の信頼性に影響を及ぼすことがあります。
- 2) すべての電圧値は回路のグランド端子を基準にしています。



本製品は、限定的なゲート保護を内蔵しています。保存時または取り扱い時は、MOS ゲートに対する静電破壊を防止するために、リード線どうしをショートさせておくか、デバイスを導電フォームに入れる必要があります。

5 電気的特性

これらの特性は、セクション 28.1 に記載されている推奨レジスタ設定を使用する場合のみ有効であることに注意してください。

5.1 推奨動作条件

パラメータ	MIN	NOM	MAX	単位
動作電源電圧	1.8		3.8	V
周囲温度	-40		125	°C

5.2 DC 特性

$T_A = 25^\circ\text{C}$ 、 $V_{DD} = 3.0\text{V}$ 、 $f_c = 2440\text{MHz}$ (特に記述のない限り)。すべてのパラメータは負荷 50Ω のテキサス・インスツルメンツ CC2520 EM 2.1 リファレンス・デザインで測定したものです。

パラメータ	条件	MIN	TYP	MAX	単位
ロジック "1" 入力電圧	すべてのパッド(GPIO および固定入力パッドの両方)に有効			80%	of VDD
ロジック "0" 入力電圧	すべてのパッド(GPIO および固定入力パッドの両方)に有効	30%			of VDD
入力パッド・ヒステリシス	RESET_N、CSn などの固定入力パッド専用		0.5		V
ロジック "0" 入力電流	入力 = 0V	-25		25	nA
ロジック "1" 入力電流	入力 = VDD	-25		25	nA

5.3 ウェークアップとタイミング

$T_A = 25^\circ\text{C}$ 、 $V_{DD} = 3.0\text{V}$ 、 $f_c = 2440\text{MHz}$ (特に記述のない限り)。すべてのパラメータは負荷 50Ω のテキサス・インスツルメンツ CC2520 EM 2.1 リファレンス・デザインで測定したものです。

パラメータ	コメント	MIN	TYP	MAX	単位
LPM2 → AM 時間	内部レギュレータ起動時間 + XOSC 起動時間		0.3		ms
LPM1 → AM 時間	XOSC 起動時間		0.2		ms
AM → RX 時間				192	μs
AM → TX 時間				192	μs
RX/TX ターンアラウンド・タイム				192	μs
TX/RX ターンアラウンド・タイム				192	μs
無線ビット・レート			250		kbps
無線チップ・レート			2.0		MChip/s

5.4 電流消費

$T_A = 25^\circ\text{C}$ 、 $V_{DD} = 3.0\text{V}$ 、 $f_c = 2440\text{MHz}$ (特に記述のない限り)。すべてのパラメータは負荷 50Ω のテキサス・インスツルメンツ CC2520 EM 2.1 リファレンス・デザインで測定したものです。

パラメータ	条件	MIN	TYP	MAX	単位
受信電流	SYNC 待ち状態		22.3	24.8	mA
	$T_A = -40 \sim 125^\circ\text{C}$ 、 $V_{DD} = 1.8 \sim 3.8\text{V}$ 、 $f_c = 2394 \sim 2507\text{MHz}$			26.3	mA
	SYNC 待ち状態、低電流 RX 設定		18.8		mA
	受信フレーム、-50dBm 入力レベル		18.5		mA
送信電流	0dBm 設定		25.8	28.8	mA
	+5dBm 設定		33.6	37.2	mA
	$T_A = -40 \sim 125^\circ\text{C}$ 、 $V_{DD} = 1.8 \sim 3.8\text{V}$ 、 $f_c = 2394 \sim 2507\text{MHz}$			37.5	mA
アクティブ・モード電流	XOSC オン、デジタル・レギュレータ・オン		1.6	1.9	mA

パラメータ	条件	MIN	TYP	MAX	単位
	$T_A = -40 \sim 125^\circ\text{C}$, $V_{DD} = 1.8 \sim 3.8\text{V}$, $f_c = 2394 \sim 2507\text{MHz}$			2.6	mA
LPM1 電流	XOSC オフ、デジタル・レギュレータ・オン 状態保持 $T_A = -40 \sim 125^\circ\text{C}$, $V_{DD} = 1.8 \sim 3.8\text{V}$, $f_c = 2394 \sim 2507\text{MHz}$		175	250	μA
LPM2 電流	XOSC オフ、デジタル・レギュレータ・オフ 保持する状態がない $T_A = -40 \sim 125^\circ\text{C}$, $V_{DD} = 1.8 \sim 3.8\text{V}$, $f_c = 2394 \sim 2507\text{MHz}$		30	120	nA
				4.5	μA

5.5 受信パラメータ

$T_A = 25^\circ\text{C}$, $V_{DD} = 3.0\text{V}$, $f_c = 2440\text{MHz}$ (特に記述のない限り)。すべてのパラメータは負荷 50Ω のテキサス・インスツルメンツ CC2520 EM 2.1 リファレンス・デザインで測定したものです。

パラメータ	条件	MIN	TYP	MAX	単位
レシーバ感度	[2]は -85dBm を要求 $T_A = -40 \sim 125^\circ\text{C}$, $V_{DD} = 1.8 \sim 3.8\text{V}$, $f_c = 2394 \sim 2507\text{MHz}$	-99	-98	-95	dBm
飽和	[2]は -20dBm を要求		6		dBm
干渉波除去	感度レベルより 3dB 高い希望波、802.15.4 チャンネルにおける 802.15.4 変調干渉波: 希望波 $\pm 5\text{MHz}$ 。[2]は 0dB を要求。 希望波 $\pm 10\text{MHz}$ 。「2」は 30dB を要求。 $\pm 20\text{MHz}$ またはそれ以上。-82dBm の希望波。		49		dB
			54		dB
			55		dB
最大スプリアス発射	30MHz ~ 1000MHz		< -80		dBm
50 Ω シングル・エンド負荷で測定。EN 300 328、EN 300 440 クラス 2、FCC CFR47 パート 15、および ARIB STD-T-66 に準拠	1GHz ~ 12.75GHz		-56		dBm
周波数許容誤差	入力レベルは感度レベルより 3dB 高い		± 400		kHz
IIP3			-24		dBm

5.6 周波数シンセサイザ・パラメータ

$T_A = 25^\circ\text{C}$, $V_{DD} = 3.0\text{V}$, $f_c = 2440\text{MHz}$ (特に記述のない限り)。すべてのパラメータは負荷 50Ω のテキサス・インスツルメンツ CC2520 EM 2.1 リファレンス・デザインで測定したものです。

パラメータ	条件	MIN	TYP	MAX	単位
位相ノイズ 無変調キャリア	キャリアから $\pm 1\text{MHz}$ でのオフセット キャリアから $\pm 2\text{MHz}$ でのオフセット キャリアから $\pm 5\text{MHz}$ でのオフセット		-111		dBc/Hz
			-118		dBc/Hz
			-128		dBc/Hz
RF 周波数範囲	1MHz ステップ・プログラマブル。準拠の 5MHz ステップを使用[2]。	2394		2507	MHz

5.6.1 送信パラメータ

$T_A = 25^\circ\text{C}$ 、 $V_{DD} = 3.0\text{V}$ 、 $f_c = 2440\text{MHz}$ (特に記述のない限り)。すべてのパラメータは負荷 50Ω のテキサス・インスツルメンツ CC2520 EM 2.1 リファレンス・デザインで測定したものです。

パラメータ	条件	MIN	TYP	MAX	単位
出力電力 注： 温度の変化による出力電力の変動を抑えるために、温度ごとに異なる設定を使用することが推奨されます。このような場合は、オンチップ温度センサを使用します。詳細については、セクション 5.11 を参照してください。	0dBm 設定	-3	1	5	dBm
	+5dBm 設定	2	5	7	dBm
	$T_A = -40 \sim 85^\circ\text{C}$ 、 $V_{DD} = 2.0 \sim 3.8\text{V}$ 、 $f_c = 2394 \sim 2507\text{MHz}$	-3		8	dBm
	$T_A = -40 \sim 85^\circ\text{C}$ 、 $V_{DD} = 1.8 \sim 3.8\text{V}$ 、 $f_c = 2394 \sim 2507\text{MHz}$	-4		8	dBm
	$T_A = -40 \sim 125^\circ\text{C}$ 、 $V_{DD} = 2.0 \sim 3.8\text{V}$ 、 $f_c = 2394 \sim 2507\text{MHz}$ $T_A = -40 \sim 125^\circ\text{C}$ 、 $V_{DD} = 2.0 \sim 3.8\text{V}$ 、 $f_c = 2394 \sim 2507\text{MHz}$	-6 -9		8 8	dBm dBm
最大出力電力での最大スプリアス発射 Texas Instruments CC2520 EM リファレンス・デザインは EN 300 328、EN 300 440、FCC CFR47 パート 15、および ARIB STDT-66 に準拠しています。 FCC の下で +5dBm における 2480MHz での送信は、デュエティ・サイクルまたは出力電力の低減に対応しています。 ピーク時のスプリアス発射では、ETSI および FCC による帯域制限 (1GHz 以下の周波数) に違反する可能性があります。放射妨害となるすべてのスプリアス発射は ETSI/FCC/ARIB の制限範囲内です。実施要件を満たす必要のあるアプリケーションでは、マッチング・ネットワークと RF コネクタ間で単純な 50 Ω ハイパス・フィルタを使用してください。	25MHz ~ 1GHz(制限帯域外)		-40		dBm
	25MHz ~ 1GHz(FCC 制限帯域内)		-53		dBm
	47 ~ 74MHz、87.5 ~ 118MHz、174 ~ 230MHz、470MHz ~ 862MHz(ETSI 制限帯域)		-42		dBm
	1800MHz ~ 1900MHz(ETSI 制限帯域)		-56		dBm
	5150MHz ~ 5300MHz(ETSI 制限帯域)		-54		dBm
	2483.5MHz 以上(FCC 制限帯域) $f_c = 2480\text{MHz}$ 、+5dBm $f_c = 2480\text{MHz}$ 、0dBm		-37 -41		dBm dBm
	2・RF および 3・RF(FCC 制限帯域)		-54		dBm

パラメータ	条件	MIN	TYP	MAX	単位
エラー・ベクトル 振幅 (EVM)	[2]は最大 35% を要求。[2] の定義に従って測定。				
	+5dBm 設定。f _c = IEEE 802.15.4 チャンネル		6		%
	0dBm 設定。f _c = IEEE 802.15.4 チャンネル		2		%

5.7 RSSI/CCA パラメータ

T_A = 25°C、VDD = 3.0V、f_c = 2440MHz(特に記述のない限り)。すべてのパラメータは負荷 50Ω のテキサス・インスツルメンツ CC2520 EM 2.1 リファレンス・デザインで測定したものです。

パラメータ	コメント	MIN	TYP	MAX	単位
RSSI 範囲			100		dB
RSSI/CCA 精度			±4		dB
RSSI/CCA オフセット	実際の RSSI = レジスタ値 - オフセット		76		dB
LSB 値			1		dB

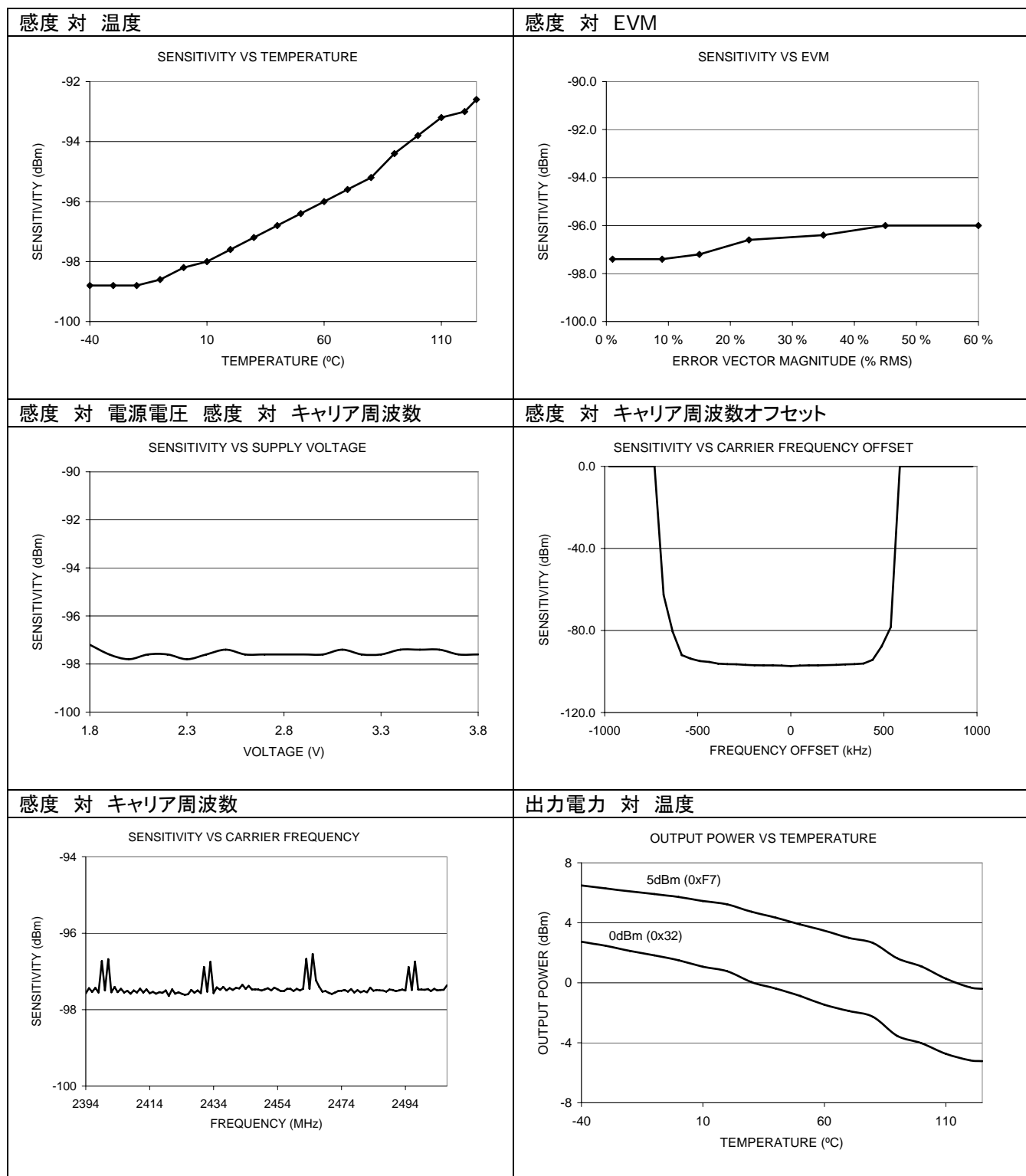
5.8 FREQEST パラメータ

T_A = 25°C、VDD = 3.0V、f_c = 2440MHz(特に記述のない限り)。すべてのパラメータは負荷 50Ω のテキサス・インスツルメンツ CC2520 EM 2.1 リファレンス・デザインで測定したものです。

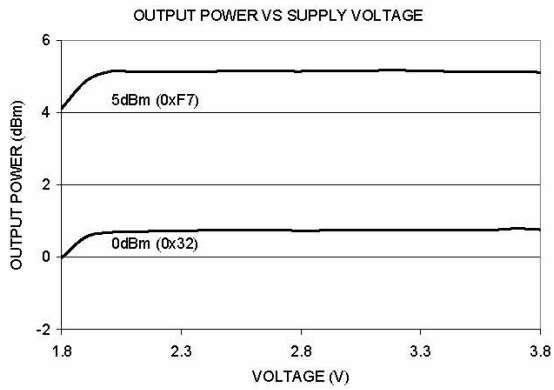
パラメータ	コメント	MIN	TYP	MAX	単位
FREQEST 範囲			±300		KHz
FREQEST 精度			±10		KHz
FREQEST オフセット	実周波数オフセット = FREQEST 値 - オフセット		64		KHz
LSB 値			7.8		KHz

5.9 標準特性曲線

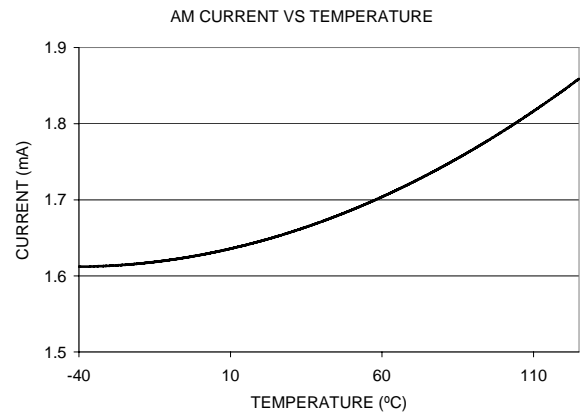
$T_A = 25^\circ\text{C}$ 、 $V_{DD} = 3.0\text{V}$ 、 $f_c = 2440\text{MHz}$ (特に記述のない限り)。すべてのパラメータは負荷 50Ω のテキサス・インスツルメンツ CC2520 EM 2.1 リファレンス・デザインで測定したものです。



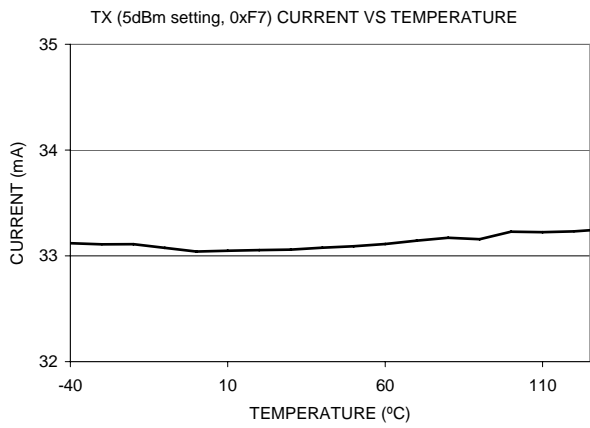
出力電力 対 電源電圧



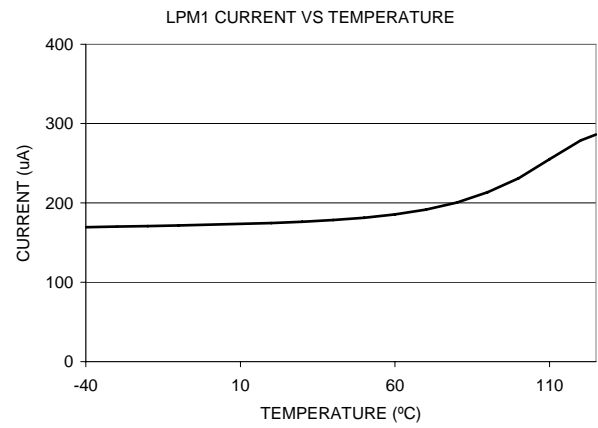
AM 電流 対 温度



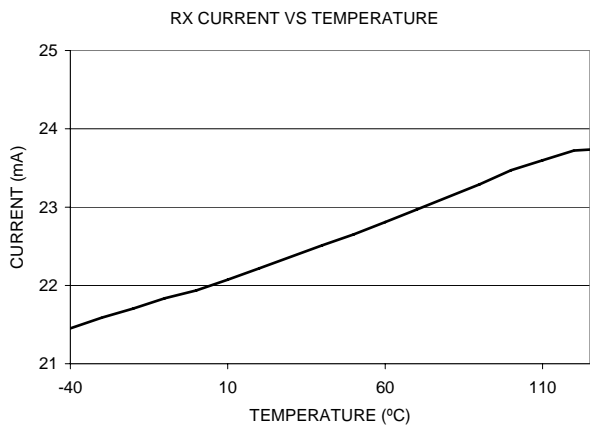
TX (5dBm 設定、0x57) 電流 対 温度



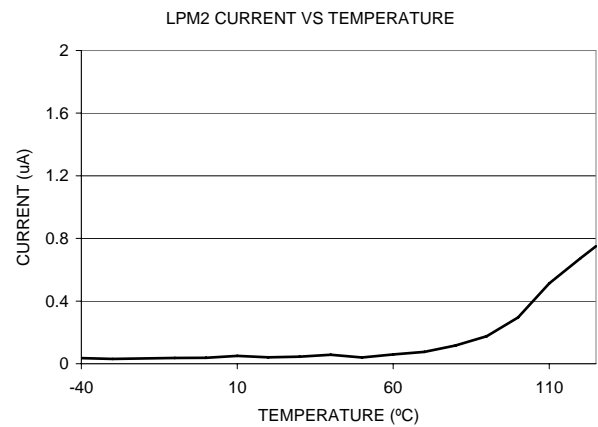
LPM1 電流 対 温度

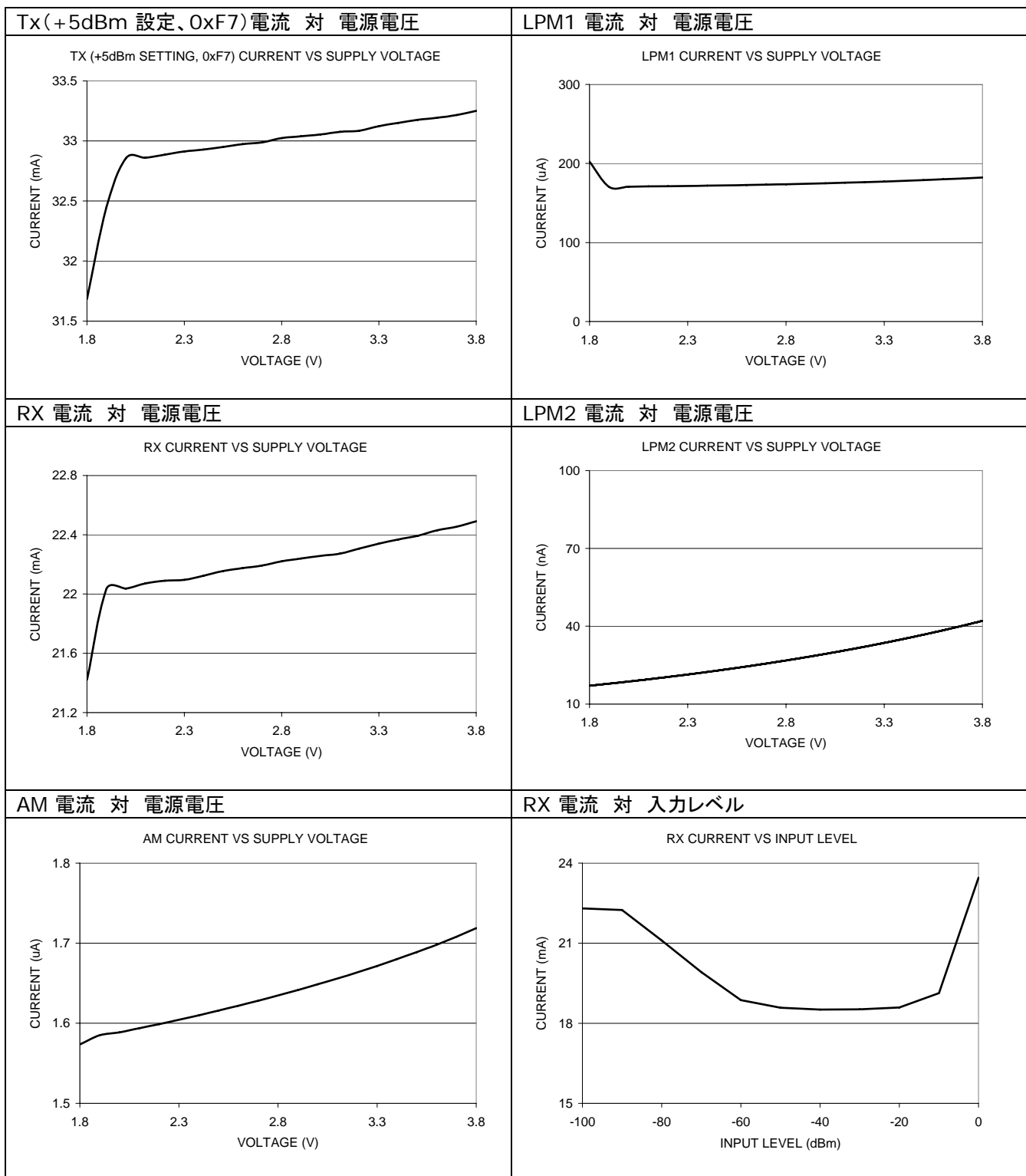


RX 電流 対 温度

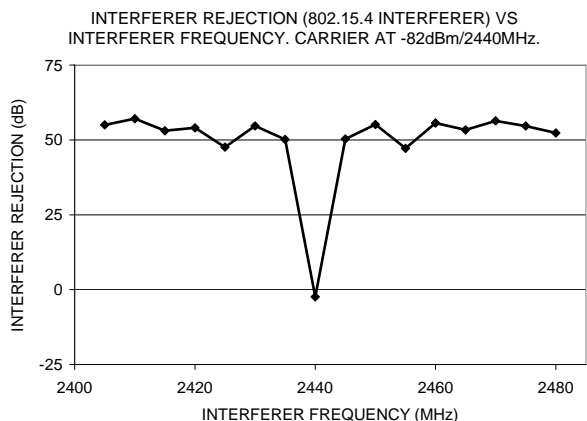


LPM2 電流 対 温度

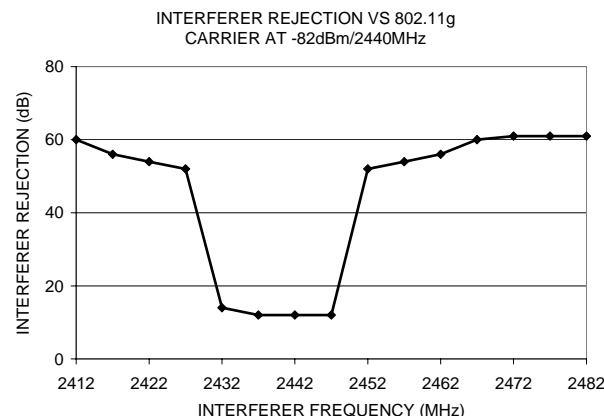




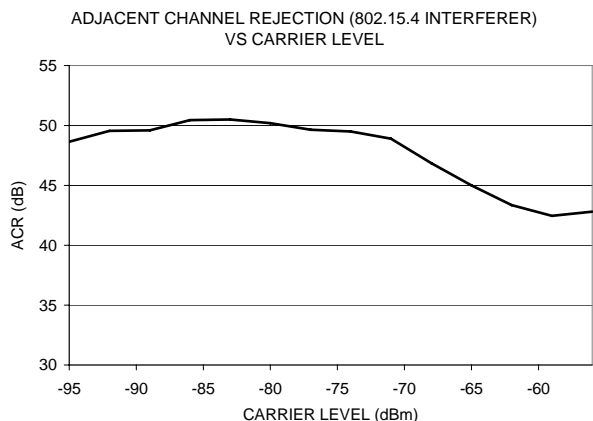
干渉源除去 (802.15.4 干渉源) 対
干渉源周波数キャリア (-82dBm/2440MHz)



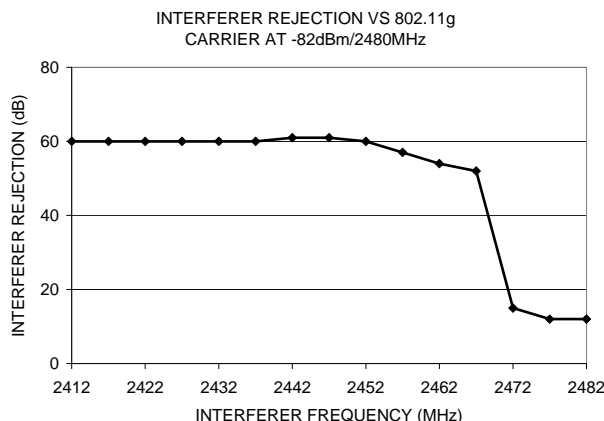
干渉源除去 対
802.11g キャリア (-82dBm/2440MHz)



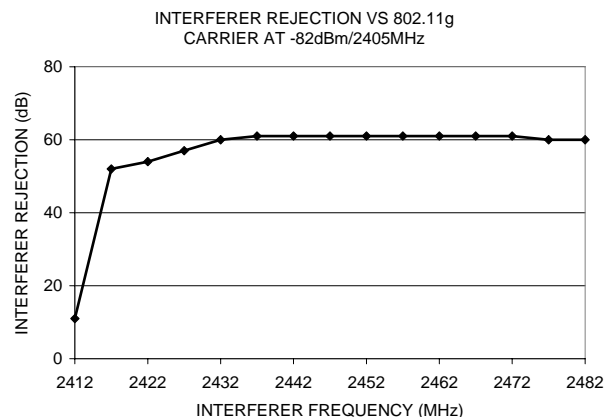
隣接チャンネル除去
(802.15.4 干渉源 対 キャリア・レベルの関係)



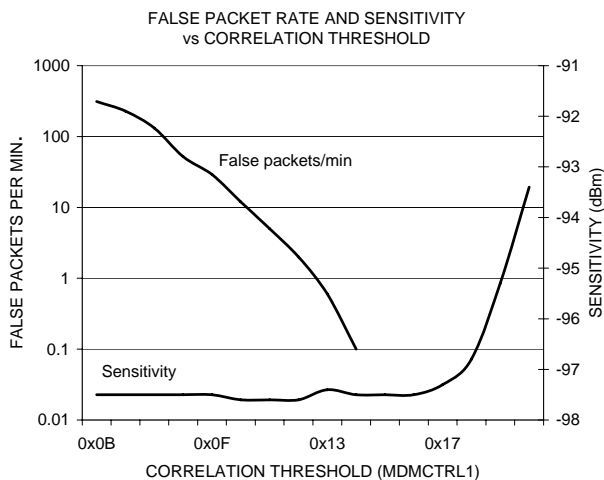
干渉源除去 対
802.11g キャリア (-82dBm/2480MHz)



干渉源除去 対
802.11g キャリア (-82dBm/2405MHz)



偽パケット・レートおよび感度 対 相関スレッショールド



<p>FREQEST 対 実際のオフセット周波数</p>	<p>温度センサ出力と電源電圧の関係 (温度 = 25°C)</p>
<p>オフセット補正 RSSI と入力レベルの関係</p>	<p>相関値と入力信号のエラー・ベクトル振幅の関係</p>
<p>温度センサ出力と温度の関係 (電源電圧 = 3V)</p>	

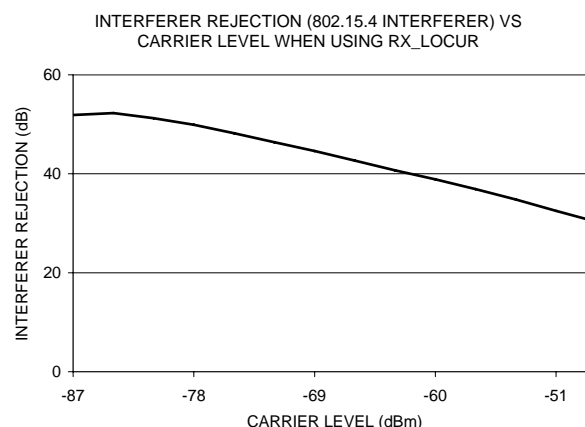
干渉源除去(802.15.4 干渉源)と
RX_LOCUR 使用時のキャリア・レベルの関係

5.10 低電流モード RX

入力信号を実際に受信するより待機する時間の方が長いアプリケーションでは、特別な低電流 RX モードを使用するとメリットが得られる場合があります。このモードでは、感度を犠牲にして電流を低減します。

このモードを使用すると、RSSI と CCA のいずれも有効にならないことに注意してください。たとえば、これらの設定は、STXONCCA と併用することはできません。また、干渉源除去は、通常の推奨設定を使用した場合と比べ、より強度の高い入力信号レベルに下がることに注意してください。

重要: 低電流 RX モードは $-40\sim 85^{\circ}\text{C}$ の範囲でのみ有効です。



5.10.1 低電流 RX モード・パラメータ

$T_A = 25^{\circ}\text{C}$ 、 $V_{DD} = 3.0\text{V}$ 、 $f_c = 2440\text{MHz}$ (特に記述のない限り)。すべてのパラメータは負荷 50Ω のテキサス・インスツルメンツ CC2520 EM 2.1 リファレンス・デザインで測定したものです。

パラメータ	条件	MIN	TYP	MAX	単位
RX 電流	SYNC 待ち		18.8		mA
感度	[2]は -85dBm を要求		-90		dBm
干渉波除去	感度レベルより 3dB 高い希望波、802.15.4 チャンネルの 802.15.4 変調干渉波:				
	希望波 $\pm 5\text{MHz}$ 。[2]は 0dB を要求。		52		dB
	希望波 $\pm 10\text{MHz}$ 。[2]は 30dB を要求。		54		dB
	$\pm 20\text{MHz}$ 超。		55		dB

表 1: 低電流 RX モード。通常の推奨設定に追加して使用した場合。

レジスタ	設定(16 進)	コメント
RXCTRL	33	感度の低下および電流消費の低減
FSCTRL	12	電流消費の低減および有効な温度範囲の縮小
AGCCTRL2	EB	感度の低下および電流消費の低減

5.11 TX のオプションの温度補償

オンチップ温度センサ(またはその他のセンサ)を使用して、実際の温度設定を調整できます。これにより、-40°C ~ 125°C の範囲での温度変化による出力電力の変動が大幅に削減されます。

このため、TX 設定は高温動作にのみ適していることが判明しています(F7125deg)。この設定は、温度が 70 度以上の場合のみ使用すべきですが、高温での出力電力低下が大幅に抑えられます。

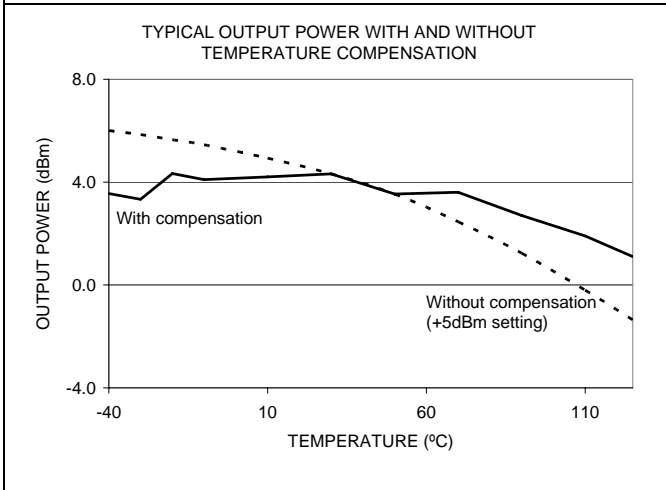
表 2: 高温動作にのみ適した F7125deg 設定(推奨設定と異なる設定のみ表示)

レジスタ	設定(16 進)	コメント
TXCTRL	94	高温での出力電力を増加
FSCTRL	7B	高温での出力電力を増加

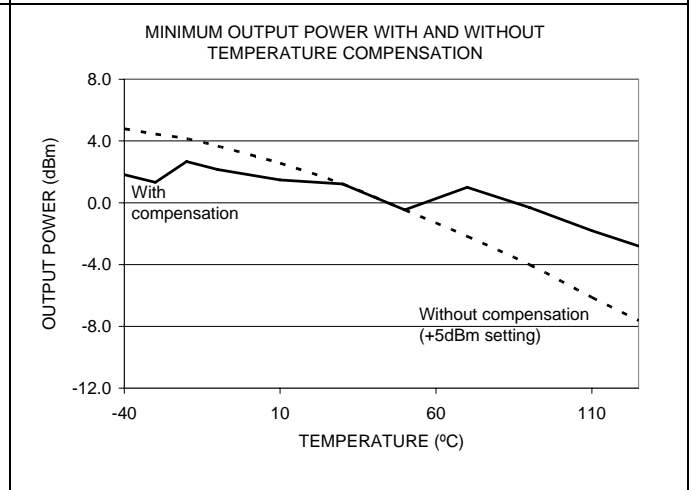
表 3: 温度に対応した推奨される TXPOWER レジスタ設定

温度	-40	-30	-20	-10	10	30	50	70	90	110	125	°C
推奨設定	13	13	AB	AB	F2	F7	F7	F7125deg	F7125deg	F7125deg	F7125deg	-
通常の出 力電力	3.6	3.3	4.3	4.1	4.2	4.3	3.5	3.6	2.7	1.9	1.1	dBm

温度補償の有無による通常の出電力の違い



温度補償の有無による最小出力電力の違い



5.11.1 温度センサの使用

オンチップ温度センサは、次の手順により、GPIO0 GPIO1 ピンを使用してアクセスできます。

- GPIOCTRL0 および GPIOCTRL1 レジスタに 0x80 を書き込んで GPIO0 および GPIO1 を入力として構成します。
- GPIOCTRL.GPIO_ACTRL='1' を設定して、これらの 2 つのピンのアナログ出力機能を有効にします。
- ATEST レジスタに 0x01 を書き込んで温度センサ出力を選択します。これにより、GPIO1 は GND を出力し、GPIO0 は温度に比例した電圧を出力するようになります。
- マイクロコントローラで ADC を使用して GPIO0 の出力電圧を測定し、温度を計算します。

温度センサからの出力は、セクション 5.9 にグラフで示されていますが、温度計算の基準として、次の数値を使用できます。

$T_c = -40 \sim 125^\circ \text{C}$ 、 $V_{DD} = 1.8\text{V} \sim 3.8\text{V}$

パラメータ	Min	Typ	Max	単位
25°C での温度センサ電圧		0.8		V
温度 センサ 出力対温度		25		mV/10°C
温度 センサ 出力対電源電圧		6		mV/V
固定電圧での温度センサ精度(キャリブレーションなし)		±12		°C
固定電圧での温度センサ精度(1 ポイント・キャリブレーション)		±1		°C

6 クリスタル固有のパラメータ

6.1 クリスタル要求

パラメータ	条件	MIN	TYP	MAX	単位
クリスタルの周波数			32		MHz
クリスタルの周波数精度要件	[2]で指定されているように初期許容誤差、エージング、および温度依存性が含まれます。オンチップ・クリスタル・チューニングを使用して緩和できます(下記を参照)。	-40		40	ppm
ESR				60	Ω
C_0				7	pF
C_L				16	pF

6.2 オンチップ・クリスタル周波数チューニング

パラメータ	条件	MIN	TYP	MAX	単位
クリスタル・チューニング・レンジ(C_{tune})	容量の追加のみ可能		7		pF
クリスタル・チューニング・ステップ・サイズ			0.4		pF
クリスタル・チューニング・ドリフト	適用されたチューニングの割合(%)		± 10		%
CC2520 EM 2.1 リファレンス・デザインを使用したクリスタル・チューニング(NX3225DA、 $C_L = 16\text{pF}$):					
起動時間			0.2		ms
クリスタル・チューニング・ステップ・サイズ	NDK クリスタル NX3225DA、 $C_L = 16\text{pF}$		3		ppm
クリスタル・チューニング・レンジ			-45		ppm
その他のクリスタルを使用したクリスタル・チューニング。数値はすべて推定値です。					
起動時間			0.2		ms
クリスタル・チューニング・ステップ・サイズ	NDK クリスタル NX4025DA、 $C_L = 13\text{pF}$		8		ppm
クリスタル・チューニング・レンジ			-120		ppm
起動時間			0.1		ms
クリスタル・チューニング・ステップ・サイズ	NDK クリスタル NX5032SA、 $C_L = 10\text{pF}$		10		ppm
クリスタル・チューニング・レンジ			-160		ppm

クリスタル・オシレータの使用の詳細については、セクション 22 を参照してください。

7 ピン配置

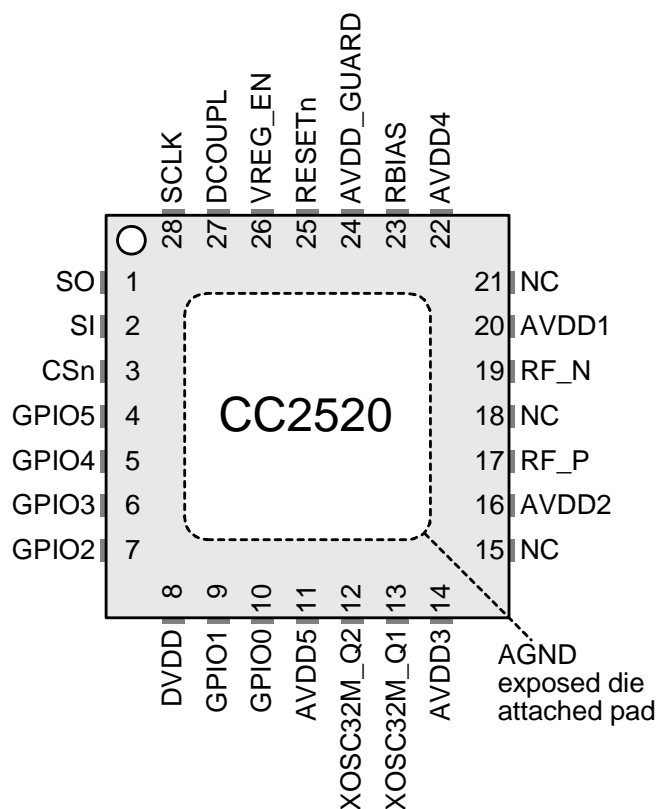


図 1: CC2520 のピン配置(上面図)

表 4: CC2520 のピン配置

信号	ピン番号	種類	概要
SPI			
SCLK	28	I	SPI インターフェイス: シリアル・クロック。最大 8MHz
SO	1	O	SPI インターフェイス: シリアル出力
SI	2	I	SPI インターフェイス: シリアル入力
CSn	3	I	SPI インターフェイス: チップ・セレクト、アクティブ "LOW"
汎用デジタル I/O			
GPIO0	10	IO	汎用 デジタル I/O
GPIO1	9	IO	汎用 デジタル I/O
GPIO2	7	IO	汎用 デジタル I/O
GPIO3	6	IO	汎用 デジタル I/O
GPIO4	5	IO	汎用 デジタル I/O
GPIO5	4	IO	汎用 デジタル I/O

信号	ピン番号	種類	概要
その他			
RESETn	25	I	外部リセット・ピン、アクティブ "LOW"
VREG_EN	26	I	高い値の場合、デジタル電圧レギュレータがアクティブになります。
NC	15,18,21		未接続
アナログ			
RBIAS	23	Analog IO	リファレンス電流の外付け高精度バイアス抵抗。56kΩ、±1%
RF_N	19	RF IO	受信モードでの LNA への負の RF 入力信号 送信モードでの PA からの負の RF 出力信号
RF_P	17	RF IO	受信モードでの LNA への正の RF 入力信号 送信モードでの PA からの正の RF 出力信号
XOSC32M_Q1	13	Analog IO	クリスタル・オシレータのピン 1
XOSC32M_Q2	12	Analog IO	クリスタル・オシレータのピン 2
電源とグラウンド			
AVDD	11,14,16,20,22	Power (Analog)	1.8V ~ 3.8V アナログ電源接続
AVDD_GUARD	24	Power (Analog)	デジタル・ノイズ・アイソレータおよびデジタル電圧レギュレータの電源接続
DCOUPPL	27	Power (Digital) O	デカップリング用 1.6V ~ 2.0V デジタル電源出力 注: このピンは、外部 デバイスの電流供給には使用できません。
DVDD	8	Power (Digital) O	デジタル・パッド用 1.8V ~ 3.8V デジタル電源
AGND	Die pad	Ground (Analog)	

8 機能概要

8.1 統合型 2.4GHz IEEE 802.15.4 準拠無線

CC2520 は、レシーバ感度が高く、干渉源に対して堅牢な、2.4GHz 帯域で動作するダイレクト・コンバージョン・トランシーバです。CC2520 無線部は、IEEE 802.15.4 PHY 仕様に準拠しています。無線には 250kbps のデータ・レートと 2M チップ/秒のチップ レートがあり、ETSI EN 300 328 および EN 300 440 クラス 2(ヨーロッパ)、FCC CFR47 パート 15(米国)、および ARIB STD-T66(日本)で定められた世界の無線周波数割り当てへの準拠を目的とするシステムに適しています。

8.2 CC2420 との比較

CC2520 では CC2420 の機能および性能が大幅に向上しました。下の表は、両方の比較を表したものです。

表 5: CC2420 と CC2520 の比較

特長	CC2420	CC2520
標準	IEEE 802.15.4-2003	IEEE 802.15.4-2006
最大出力電力	0dB	+5dB
標準感度	-95dBm	-98dBm
汎用クロック出力	なし	あり。構成可能な周波数 1MHz ~ 16MHz
ユーザ・インターフェイス	コマンド・ストローブおよびコンフィギュレーション・レジスタ。ユーザ・コントロールはすべて SPI を通過します。	命令セット(サブセットとしてコマンド・ストローブを含む)およびコンフィギュレーション・レジスタ。コマンド・ストローブは、タイミング・コントロールが GPIO ピンによってトリガできます。ステータス情報が改善されました。
レジスタ・アクセス	クリスタル・オシレータが動作していなくてもアクセス可能	クリスタル・オシレータの動作中のみアクセス可能
デジタル入力	シュミット・トリガなし	すべてのデジタル入力でのシュミット・トリガ
デジタル出力	固定構成	柔軟性が高く、構成が可能
起動	XOSC の手動スタート	XOSC はリセット(reset_n pin)後に自動的に起動します。SRES 命令実行後の XOSC の手動スタート
クリスタルの周波数	16MHz	32MHz
パケット・スニフing	ハードウェア・サポートなし	送受信フレームへの介入がないスニフingをハードウェアでサポート
最大 SPI クロック速度	10MHz	8MHz
RAM サイズ	364 バイト	768 バイト
動作電圧	2.1V ~ 3.6V	1.8V ~ 3.8V
最大動作温度	85°C	125°C

特長	CC2420	CC2520
セキュリティ	限定された柔軟性	柔軟性の高いセキュリティ命令。 RAM サイズの増加により柔軟性の高い処理が可能
パッケージ	QLP-48、7mm x 7mm	QFN 28(RHD)、5mm x 5mm
RF 周波数範囲	2400MHz ~ 2483.5MHz	2394MHz ~ 2507MHz

8.3 ブロック図

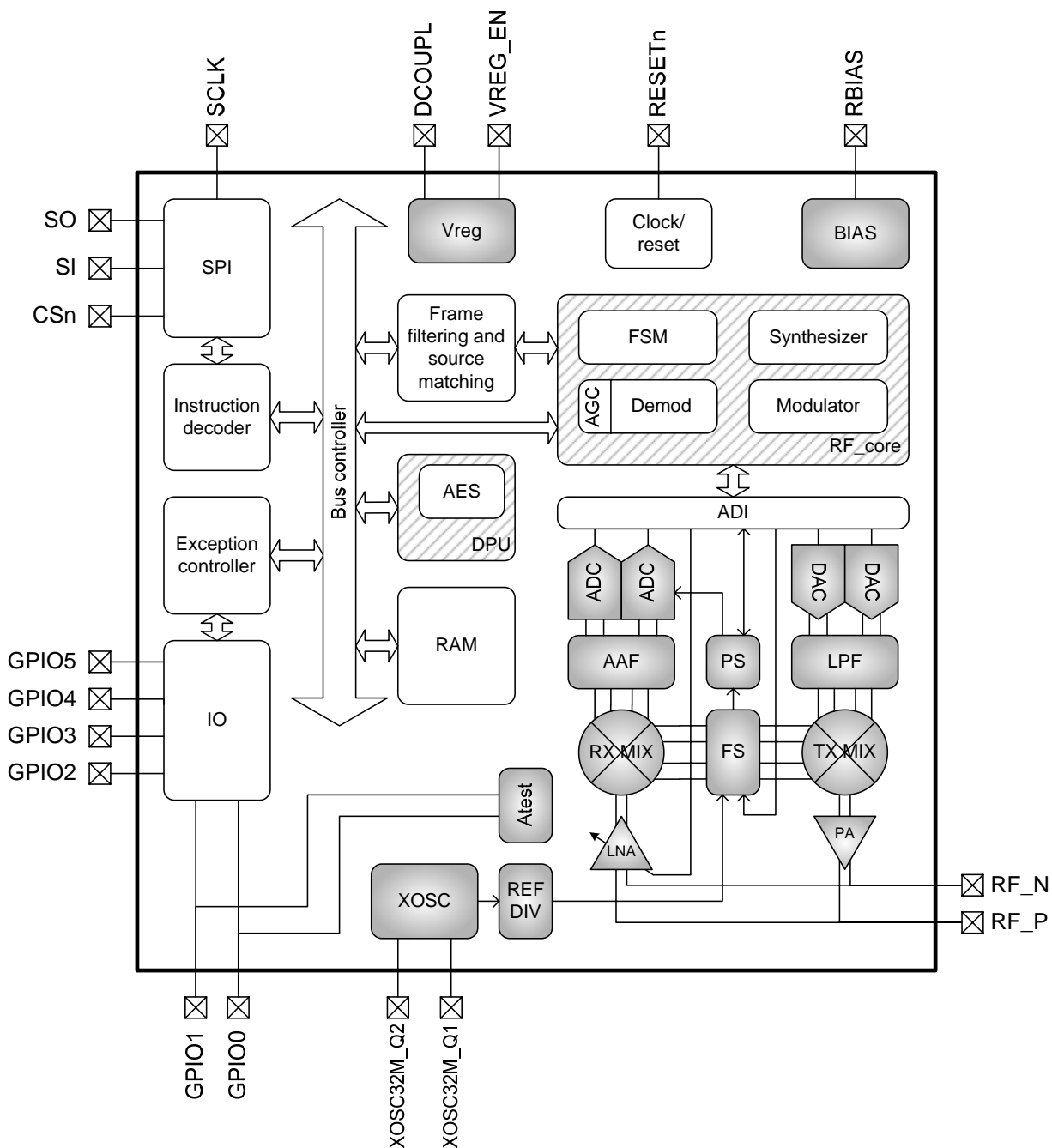


図 2: CC2520 ブロック図

CC2520 は通常、**SPI** および一部の GPIO に接続されたマイクロコントローラによって制御されます。マイクロコントローラは CC2520 に命令を送信し、命令は**インストラクション・デコーダ**によって実行されるか、別のモジュールに引き渡されます。

命令または外部イベント(フレームの受信など)の実行により、1 つまたは複数の例外が発生する可能性があります。例外は、タスクを自動化するためのきわめて柔軟性の高いメカニズムです。たとえば、別の命令の実行のトリガに使用、または GPIO ピンに送信してマイクロコントローラに対する割り込み信号として使用することもできます。**例外コントローラ**は、例外処理を実行します。

マイクロコントローラは通常、1 つまたは複数の GPIO ピンに接続されます。各ピンの機能は、レジスタ設定に基づいて **IO** モジュールによって個別に制御されます。GPIO ピンでは大量の内部信号を監視できます。また、GPIO ピンは、入力として構成して、特定の命令の実行をトリガするのに使用することもできます。これは通常、マイクロコントローラが命令のタイミングを正確に制御する必要がある場合に使用されます。

RAM モジュールには、(固定のアドレス範囲での) FIFO の送受信および別のデータの一時的なストレージとして使用されるメモリが含まれています。汎用メモリ・アクセスおよび FIFO アクセスには、個別の命令があります。

データ処理装置(DPU)は、より高度な命令の実行を処理します。DPU には、セキュリティ命令の実行中に使用される AES コアが含まれています。メモリ管理(コピー、インクリメントなど)も DPU によって実行されます。

クロック/リセット・モジュールは、内部クロックおよびリセット信号を生成します。

RF コアには、アナログ無線モジュールをサポートおよび制御する複数のサブモジュールが含まれています。

FSM サブモジュールは、RF トランシーバの状態、トランスミッタとレシーバの FIFO、およびアナログ・モジュールのパワー・アップやパワー・ダウンなどの動的に制御されるアナログ信号を制御します。FSM は、イベントの正しいシーケンス(レシーバを有効にする前に FS キャリブレーションを実行するなど)を制御するのに使用されます。また、復調器からの受信フレームを段階的に(フレーム長の読み取り、受信済みバイト数のカウント、FCS の確認、オプションでフレーム受信後の ACK フレームの自動送信の処理)処理します。TX でも、オプションで送信前に CCA を実行し、送信後に RX に自動的に移動して ACK フレームを受信するなどの、同様のタスクを実行します。最後に、FSM は変調器/復調器と RAM の TXFIFO/RXFIFO 間のデータの転送を制御します。

変調器は、未加工データをトランスミッタ DAC への I/Q 信号に変換します。この変換は、IEEE 802.15.4 基準に従って行われます。

復調器は、受信信号から送信データを取得します。

復調器からの振幅情報は、**自動ゲイン制御(AGC)**で使用されます。AGC は、レシーバ内の信号レベルがおおむね一定になるよう、アナログ LNA のゲインを調整します。

フレーム・フィルタリングおよびソース・マッチングは、IEEE 802.15.4 で定義されたフレーム・フィルタリングおよびソース・アドレス・マッチングの実行に必要なすべての動作を実行することで、RF_core 内の FSM をサポートします。

xosc モジュールは、XOSC32M_Q1 ピンおよび XOSC32M_Q2 ピンに接続されたクリスタルのインターフェイスとして動作します。xosc モジュールは、デジタル部および RF システムのクロックを生成し、プログラマブル・クリスタル周波数チューニングを実装します。

バイアス・モジュールは、電圧および電流のリファレンスを生成します。このモジュールは、図 3 のアプリケーション回路に示されている高精度(1%)の 56k Ω 外部抵抗に依存しています。

TX DAC は、デジタル・ベースバンド信号をアナログ信号に変換します。

LPF を通過後、信号はアップ・コンバータの複素ミキサである **TXMIX** モジュールにフィードされます。

PA は、TX の実行中に RF 信号を最大 -5dBm まで増幅します。

LNA は、受信した RF 信号を増幅します。ゲインは、デジタル AGC モジュールによって制御されるため、感度および干渉源除去が最適化されます。

RXMIX モジュールは、信号をベースバンド信号に変換する複素ダウンミキサです。

パッシブなアンチエイリアシング・フィルタ(AAF)は、ダウン・ミキシングの後で信号にローパス・フィルタを適用します。

ローパス・フィルタは I 信号と Q 信号に適用され、**ADC** によってデジタル化されます。

周波数シンセサイザ(FS)は、RF 信号のキャリア波を生成します。

電圧レギュレータ(Vreg)は、 1.8V の電源電圧をデジタル・コアに供給します。電圧レギュレータには、 $\sim 32\text{mA}$ 以上の電流で有効な電流リミッタが含まれています。

9 アプリケーション回路

CC2520 の動作に必要な外部コンポーネントはごくわずかです。一般的なアプリケーション回路については、Figure 4 を参照してください。ただし、ボード・レイアウトは記載されていないことに注意してください。ボード・レイアウトは、CC2520 の RF 性能に大きく影響します。

このセクションでは、概要のみを説明しています。詳細については、ボード・レイアウトの一覧およびメーカーと型番に対応した部品表が記載されたリファレンス・デザインを参照してください。リファレンス・デザインは、CC2520 プロダクト・フォルダ[7]からダウンロードできます。

デカップリング・コンデンサは、下の図には含まれていないことに注意してください。完全な部品表については、リファレンス・デザインを参照してください。

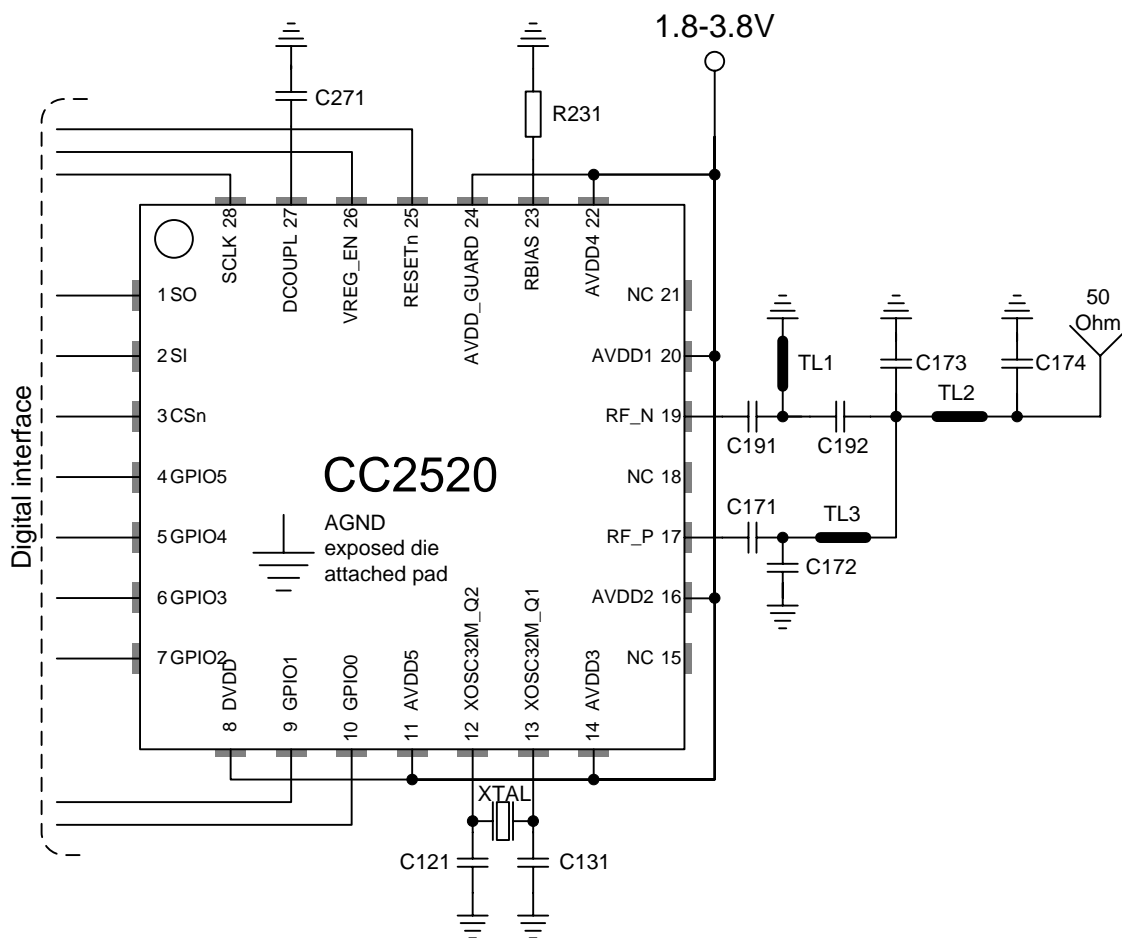


図 3: シングルエンド動作のマイクロ・ストリップライン・バランが含まれた一般的なアプリケーション回路

低コストの小型アンテナの詳細については、アンテナ・セレクション・ガイド[12]を参照してください。

9.1 入力/出力マッチング

RF 入力/出力は、ハイ・インピーダンスの差動です。

モノポールなどの不平衡アンテナを使用する場合は、性能を最適化するためにバランを使用する必要があります。バランは、低コストのディスクリット・インダクタとコンデンサのみを使用するか、ディスクリット・インダクタに代わる送信ラインと組み合わせることで実装できます。

Figure 4 は、2 層リファレンス・デザインで実装されたバランを表しています。このバランは、3 つのストリップライン(L1、L2、および L3)とディスクリット・コンポーネント(C191、C171、C192、C173、および C174)で構成されています。この回路は、アンテナ接続が 50Ω の負荷となる CC2520 に対して最適な RF ターミネーションを提供します。

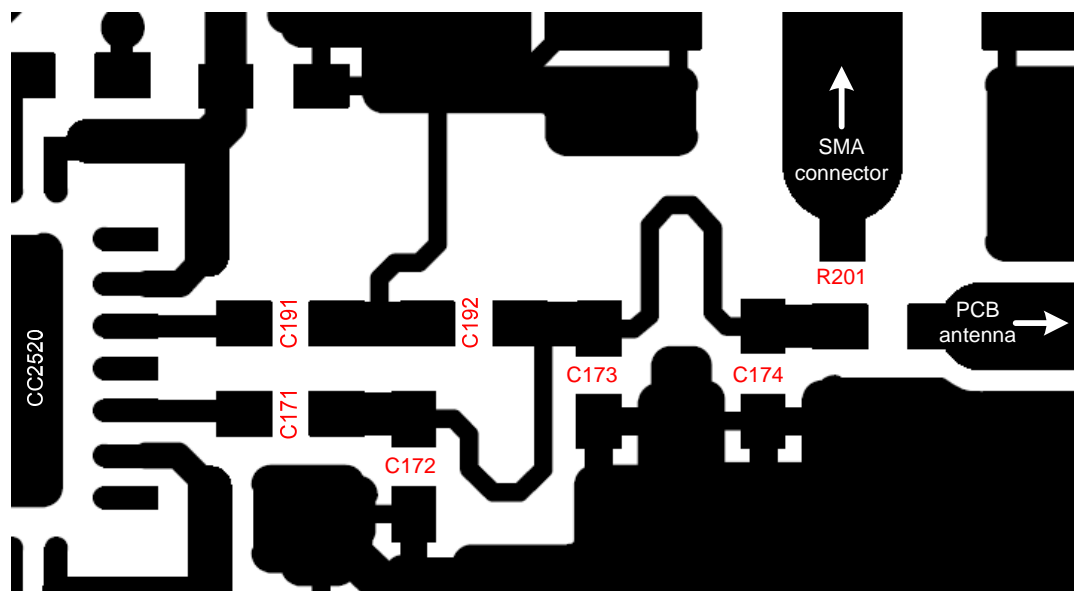


図 4: リファレンス・デザイン(Rev.2.1)の RF セクションの実際のボード・レイアウト

9.2 バイアス抵抗

バイアス抵抗 R231 は、正確なバイアス電流を設定するのに使用されます。高精度($\pm 1\%$)の $56k\Omega$ 抵抗を使用する必要があります。

9.3 水晶振動子

外付け水晶振動子(32MHz)と 2 個の負荷コンデンサ(C121 および C131)が水晶発振器に使用されます。

シングルエンド信号を XOSC32M_Q1 ピンに入力することが可能です、その場合クリスタルは使用しません。

9.4 デジタル電圧レギュレータ

オンチップ電圧レギュレータは、1.8V の電圧を CC2520 のデジタル部に供給します。C271 は電圧レギュレータのデカップリング・コンデンサです。その他の IC への電源供給にデジタル電圧レギュレータを使用することはできないことに注意してください。

9.5 電源のデカップリングおよびフィルタリング

最適な性能を実現するためには、適切な電源デカップリングが必要です。これは、一まとめにしてコンデンサ C1 として Figure 4 に示されています。デカップリング・コンデンサと電源フィルタリングの配置およびサイズは、アプリケーションの最適な性能を実現するために非常に重要です。TI では、参考にすべき便利なリファレンス・デザインを用意しています。

9.6 ボード・レイアウトのガイドライン

リファレンス・デザイン[5]からボード・レイアウトをコピーすることを推奨します。

- CC2520 の電源にはスター・ポロジの使用を推奨します。

- 電源のデカップリング・コンデンサ C1 は一まとめの部品です。実際のボード・レイアウトでは、各電源ピンにできるだけ近い場所に個別のデカップリング・コンデンサがあります。
- バランはレイアウトをかなり重視します。Figure 4 のインダクタは、実際は PCB に組み込まれたストリップラインで、ボード・レイアウトに従って値を調整する必要があります。また、コンデンサ C192、C172、C173、および C174 の値も実際のボード・レイアウトに適合させる必要があります。
- GPIO ピンは、内部プルアップ抵抗を使用するよう構成できます。リセット後または LPM2 では有効になりません。MCU が構成を変更できるようになるには時間がかかるため、これらの信号を接続する場合は、デフォルトの GPIO 構成を検討してください。LPM2 GPIO5(入力として構成)は、グラウンドまたは VDD のどちらかに接続する必要があります。その他の GPIO ピンはグラウンドに接続するか、インピーダンスを高くする必要があります。これを行わないと、電流消費は必要以上にきわめて高くなります。
- CSn が "HIGH" か、デバイスがリセットまたは LPM2 にある場合、SO ピンは、入力として構成されます。これにより、複数の SPI スレーブを 1 つの SPI マスタに接続できるようになります。必要以上に電流を消費するので、このピンは、LPM2 ではフローティングのままにしないようにします。電圧レベルを制御する方法が他にない場合は、1MΩ プルダウン抵抗を使用してください。
- 水晶の入カラインは、互いに可能な限り離して接続する必要があります。
- NC ピンは、フローティングのままにできます。
- デジタル入力のグリッチによって、システム設計に深刻な問題が生じる場合があります。デジタル入力パッドには、グリッチの影響をあまり受けないようにするためのシュミット・トリガがありますが、ボード・レイアウトではデジタル入カラインを別のノイズ信号に接続しないようにする必要があります。

9.7 アンテナの考慮事項

リファレンス・デザインには、2 つのアンテナ・オプションがあります。デフォルトでは、SMA コネクタは 0Ω 抵抗でバランに接続されます。この抵抗の半田付けをはずして時計回りに 90° 回転させ、PCB アンテナに接続すると、プレーナ型の反転された F アンテナ(PIFA)になります。

テストおよび特性評価はすべて、SMA コネクタを使用して実施されていることに注意してください。PCB アンテナについては、2 つの EM 間にリンクを確立することで、機能テストのみ実施されています。この実験では、SMA コネクタに接続されたアンテナを使用した場合とほぼ同じ範囲で PCB アンテナを使用できました。

詳細については、アンテナ・セレクション・ガイド[12]および反転型 F アンテナのアプリケーション・ノート[11]を参照してください。

9.8 マイクロコントローラとの最適な相互接続の選択

- 4 つの SPI 信号(CSn、SCLK、SI、および SO)をマイクロコントローラに接続します。これらの信号は、CC2520 を構成して、データを交換するのに必要です。
- RESETn をマイクロコントローラに接続します。たとえば、電源投入後に CC2520 をリセットする場合は、RESETn 信号を使用する方法が推奨されます。ピンを節約する必要がある場合は、RESETn ピンを使用して VDD に接続できます。CC2520 は、SRES コマンド・ストロブを使用してリセットすることもできます。この場合も SXOSCON コマンド・ストロブを発行して、クリスタル・オシレータを手動で開始する必要があります。
- VREG_EN をマイクロコントローラに接続すると、CC2520 を LPM2 に入れて節電できます。VREG_EN は VDD に接続できるため、レギュレータの電源は常にオンにしておきます。対象のアプリケーションで節電が重要でない場合、これはピンを節約するのに適した方法です。
- 1 つまたは複数の GPIO をマイクロコントローラに接続するのはオプションとなります。接続する GPIO の数は、アプリケーションによって異なります。さらに多くの GPIO をマイクロコントローラに接続すると、GPIO を個別に再構成する必要性が減るため、一般に柔軟性が高くなり、SPI トラフィックが少なくなります。
- CC2520 でマイクロコントローラにクロックを供給する場合は、GPIO0 をマイクロコントローラのクロック入力に接続する必要があります。リセットが完了すると、GPIO0 は 1MHz のクロック信号を 50/50 デューティ・サイクルで出力します。

CC2520 のデジタル IO の詳細については、セクション 12 を参照してください。

9.9 CC2520 と MSP430F2618 とのインターフェイス

MSP430F2618 は CC2520 での使用に適しています。これら 2 つのチップのインターフェイスについては、Table 5 を参照してください。表 6 に示されている相互接続は、CC2520 開発キットエラー! 参照元が見つかりません。で使用されているのとまったく同じです。

表 6 : MSP430F2618 と CC2520 の相互接続

CC2520	MSP430F2618
VREG_EN	P01.0/TACLK/CAOUT
RESETn	P05.7/TBOUTH/SVSOUT
SCLK	P05.3/UCB1CLK/UCA1STE
SO	P05.2/UCB1SOMI/UCB1SCL
SI	P05.1/UCB1SIMO/UCB1SDA
CSn	P05.0/UCB1STE/UCA1CLK
GPIO0	P01.3/TA2
GPIO1	P01.5/TA0
GPIO2	P01.6/TA1
GPIO3	P01.1/TA0/BSLTX
GPIO4	P01.2/TA1
GPIO5	P01.7/TA2

MSP430F2618 と CC2520 の相互接続を表した簡単な図は Figure 8 にあります。MSP430F2618 の詳細については、[10]を参照してください。

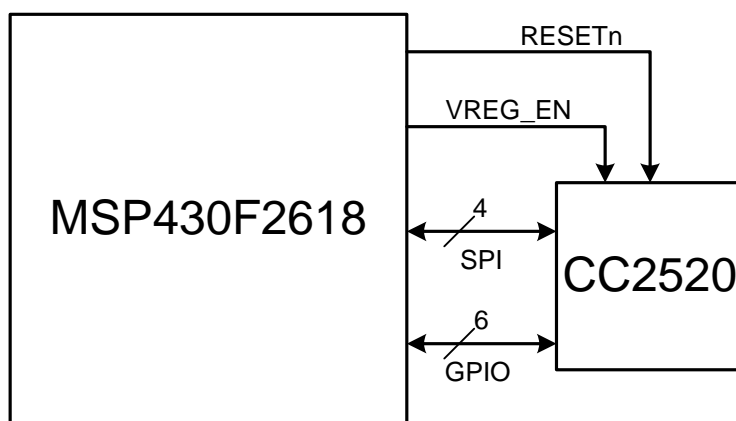


図 5: MSP430F2618 と CC2520 の相互接続

10 シリアル・ペリフェラル・インターフェイス(SPI)

SPI は CC2520 に命令を送信し、CC2520 とマイクロコントローラの間でデータを転送するためのインターフェイスです。CC2520 4 線式スレーブ・インターフェイスは、3 つの入力信号(CSn、SCLK、SI)および 1 つの出力信号(SO)から構成されます。

セクション 15 には、SPI インターフェイスを介して使用できるすべての命令の一覧と説明が記載されています。命令はバイト・オリエンテッドで、インターフェイスを介して CC2520 へ 1 バイト以上の送信が必要です。1 バイトを転送するには、CSn を "LOW" に引き下げ、SCLK でポジティブ・エッジで始まる 8 つの期間を完了する必要があります。SCLK の最長期間の要求または、連続である必要はありません。CSn が "LOW" に保たれている限り、SCLK はいつでも停止でき、必要に応じて再び開始できます。

10.1 CSn

CSn は SPI 用の入力イネーブル信号で、外部 MCU によって制御されます。CSn 信号は SPI モジュールに対する非同期型のアクティブ "HIGH" リセットとして使用されます。

CSn は SPI のすべての処理で "LOW" にしておく必要があり、SCLK の最初のポジティブ・エッジの前の 3 つ以上の期間の XOSC でも "LOW" にしておく必要があり、SCLK の最後のネガティブ・エッジの後の 3 つ以上の期間の XOSC でも "LOW" にしておく必要があります。

CSn が "HIGH" の場合、2 つ以上の期間の XOSC で "HIGH" に保っておく必要があります。

CSn は、実行された最後の命令に一定数のバイトが含まれる場合のすべての SPI 動作で "LOW" にしておくことができますが、命令コントローラの一部は実行中になるため、不要な電力を消費することになります。

一定数のバイトが含まれた命令は、セクション 15.3 の命令の一覧表に記載されています。SRXON(1 バイト)および RXMASKAND(3 バイト)には定数のバイト数が含まれており、REGRD(2 バイト以上)にはユーザによってコントロールされるバイト数が含まれています。後者は表中で命令コマンドの最後に必要なバイト(REGRD 用のバイト 3)の後にあるバイト列に 3 つのピリオド(...)が表示されています。

ユーザによってコントロールされるバイト数が含まれた命令は、立ち上がり CSn によって終了します。

ステータスは、すべての命令の最初のバイトを実行中に SO の最初のバイトとして出力されます。CSn を立ち上げずに連続的に 2 つの命令を転送すると、SO のステータス・バイトには正しい電流ステータスが含まれていない場合があります。ただし、命令の 2 番目のバイトのステータスは更新されます。2 番目の命令バイトの実行中にステータスを出力する RXMASKAND は、2 番目のバイトの実行中に正しいステータスを出力します。

電源投入後に CSn を "LOW" に引き下げると、SO は内部の XOSC 安定信号を組み合わせるため、XOSC 安定状況を見つけるのに SCLK のエッジが必要になります。CSn を "LOW" に引き下げて SO が "LOW" の場合、XOSC はまだ安定していないため、デジタル部にはクロックがありません。電源投入から XOSC までに安定させる必要のある最大時間については、セクション 5.3 で説明しています。

10.2 SCLK

SCLK は外部 MCU によって制御される、CC2520 への入力クロックです。SCLK は CC2520 の内部 XOSC クロックとは非同期です。SCLK の最大周波数は 8MHz です。最小周波数要件はありません。

10.3 SI

SI はマイクロコントローラから CC2520 へのシリアル・データ入力です。データは MSB の先頭(命令コマンドの各バイトのビット 7)を使用して送信する必要があります。

データは SCLK のネガティブ・エッジにセットアップする必要があり、SCLK の次のポジティブ・エッジによって CC2520 にクロック入力されます。

10.4 SO

SO は CC2520 から外部 MCU に送信されるシリアル・データです。データは SCLK のネガティブ・エッジでシリアル出力されるため、SO 信号を以下の SCLK 立ち上がりエッジでサンプリングする必要があります。MSB(レジスタ定義のビット 7)は、最初にシリアル出力されます。

CSn が "HIGH" の場合または RESETn が "LOW" の場合、SO は入力として構成されます。必要以上に電流消費が多くなるため、LPM1 または LPM2 内の SO ピンはフローティングのままにしないでください。

10.5 SPI タイミング要求

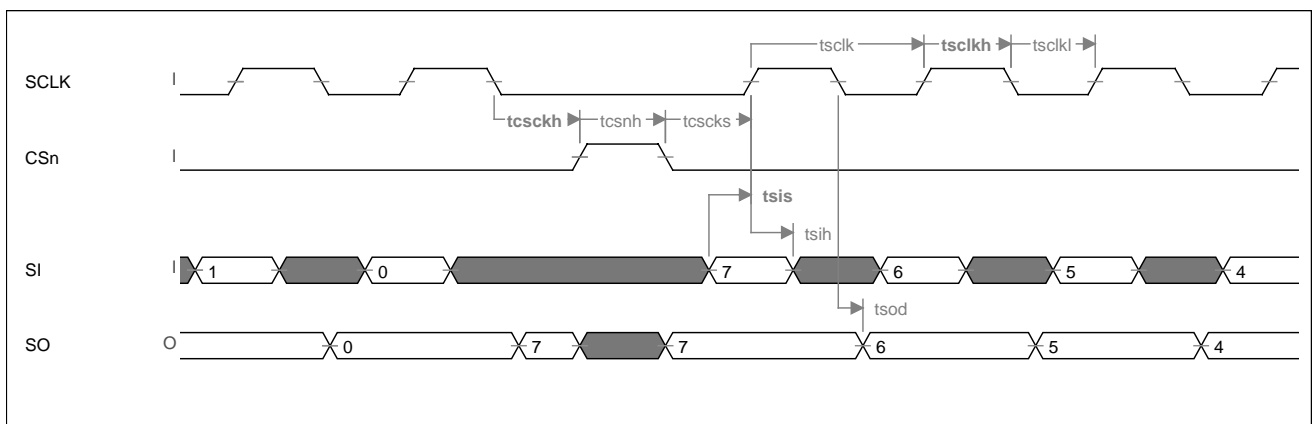


図 6: SPI タイミングの関係

次の表と図は、外部マイクロコントローラと CC2520 の SPI インターフェイスの間に必要なタイミング関係を表しています。

表 7: SPI タイミング要求

パラメータ	概要	MIN	TYP	MAX	単位
tcscks	CSn から SCLK へのセットアップ時間	62.5			ns
tcsckh	SCLK から CSn へのホールド時間	62.5			ns
tcsnh	CSn HIGH	62.5			ns
tsclk	SCLK 期間	125			ns
tsclkh	SCLK HIGH 時間	62.5			ns
tsckl	SCLK LOW 時間	62.5			ns
tsis	SI から SCLK へのセットアップ時間	31			ns
tsih	SI から SCLK へのホールド時間	31			ns
tsod	SCLK から SO への遅延			31	ns

11 GPIO

CC2520 には入力、出力、およびプルアップ抵抗として個別に構成できる 6 個の GPIO ピンがあります。MSB がピンを入力または出力に構成すると、各 GPIO はレジスタ GPIOCTRLn に関連付けられます。GPIOCTRL レジスタは、各 GPIO ピンのプルアップ、すべてのピンの追加ドライブ、およびピン 0 と 1 のアナログ機能を制御します。GPIO のテスト機能および監視機能の詳細については、セクション 30 を参照してください。

LPM2 で入力として構成されている GPIO5 は、LPM2 の入力時にグランドまたは VDD に接続されている必要があることに注意してください。GPIO5(またはその他の入力)がフローティングのままの場合、予想外の電流消費が発生します。

11.1 GPIO ピンのリセット構成

GPIO ピンのリセット設定は、下の表に示されています。これは、デバイスが LPM2 内にある場合に使用される構成でもあります。別の GPIO 設定が必要な場合は、CC2520 が LPM2 に入るたびに GPIO を再構成する必要があります。

この特別なリセット構成は、CC2520 ができる限り CC2420 と同じになるよう設定されています。

表 8: GPIO リセット状態

GPIO ピン	方向	値	プルアップ	追加ドライブ	極性	信号	GPIOCTRLn 値(16 進)	概要
0	出力	0	なし	なし	正	クロック	0x00	50/50 デューティ・サイクルの 1MHz クロック信号
1	出力	0	なし	なし	正	FIFO	0x27	1 つまたは複数のバイトが RX FIFO にある場合は "HIGH" になります。RX FIFO オーバーフローが発生した場合は "LOW" になります。
2	出力	0	なし	なし	正	FIFOP	0x28	RX FIFO 内のバイト数がプログラマブル・スレッシュホールドを超えるか、RX FIFO 内に完全フレームが 1 つ以上ある場合は "HIGH" になります。RX FIFO オーバーフローが発生した場合も "HIGH" になります。
3	出力	0	なし	なし	正	CCA	0x29	クリア・チャンネル・アセスメント。この信号の動作を構成する方法については、FSMSTAT1 レジスタを参照してください。
4	出力	0	なし	なし	正	SFD	0x2A	SFD が受信済みまたは送信済みの場合、ピンは "HIGH" になります。RX/TX を終了するたびにクリアされます。
5	入力	グランドまたは VDD に接続	なし	なし	正		0x90	機能しない

11.2 入力としての GPIO

入力として構成すると、GPIO ピンを使用して 16 個のコマンド・ストローブ(セクション 15 を参照)のうち 1 つをセクション 12.6 の GPIO 構成表に示されているようにトリガできます。これらのコマンド・ストローブは、使用可能なすべての SPI 命令のサブセットです。コマンド・ストローブは、GPIOPOLARITY レジスタの設定に応じて、立ち上がりエッジまたは立ち下がりエッジを GPIO ピンに適用することでトリガされます。ピンによってトリガされるコマンド・ストローブは、GPIOCTRLn の 7LSB によって設定されます。

例: 立ち上がりエッジで SACK 命令を実行するよう GPIO2 をセットアップします。

- GPIOPOLARITY[2] を '1' に設定します。GPIO ピン 2 は、アクティブな立ち上がりエッジに設定されます。
- GPIOCTRL2[7:0] を "1000 0101" に設定します。GPIO ピン 2 はこれで入力になり、SACK 命令に接続されます。

11.3 出力としての GPIO

GPIO ピンを出力として構成した場合、信号は GPIOCTRLn レジスタの CTRLn 設定に対応します(CTRLn 値はセクション Table 8 の 12.6 を参照)。ピンの極性は、GPIOPOLARITY レジスタに設定されます。

例: アクティブ "HIGH" レベルを示す sniff_data を出力するよう GPIO3 をセットアップします。

- GPIOPOLARITY[3] を '1' に設定します。GPIO ピン 3 はアクティブ "HIGH" レベルを示すよう設定されます。
- GPIOCTRL3[7:0] を "0011 0010" に設定します。GPIO ピン 3 はこれで出力になり、sniff_data を出力します。

11.4 GPIO の方向の切り替え

出力から入力に切り替える場合は、コマンド・ストローブをトリガしてしまうことがないように十分に注意する必要があります。入力をトリガするコマンド・ストローブ(Table 8 の最初の 16 個のエントリのうち 1 つ)に GPIOn を変更する場合は、次の手順を実行して、ピンが "HIGH" の間は方向が変更されないようにする必要があります。

1. 0x7E を GPIOCTRLn に書き込んで、定数 0 を出力するようにします。
2. '0' をマイクロコントローラから GPIO ピンに移動します。
3. たとえば、0x88 を GPIOCTRLn に書き込んで、STXON コマンド・ストローブをトリガする入力に変更します。

11.5 GPIO の構成

Table 8 では、GPIO ピンで出力として使用できる信号について説明します。CTRLn 列には、説明した機能を取得するために GPIOCTRL0-GPIOCTRL5 レジスタの 1 つに書き込む必要がある構成値が表示されます。Table 8 の IN 列には、GPIO が入力として構成され、(正しい極性の)エッジが適用されている場合に実行されるコマンド・ストローブが表示されます。OUT 列には、GPIO が出力として構成されている場合に、ピンで監視できる内部信号の名前が表示されます。

表 9: GPIO の構成

CTRLn (16 進)	IN(コマンド・ストローブ)	OUT	OUT 信号の説明
0x00	SIBUFEX	Clock	クロック信号。1MHz ~ 16MHz のプログラマブル周波数。
0x01	SRXMASKBITCLR	RF_IDLE	RF_IDLE 例外。詳細については、Table 14: Exceptions summaryを参照してください。
0x02	SRXMASKBITSET	TX_FRM_DONE	TX_FRM_DONE 例外。詳細については、Table 14: Exceptions summaryを参照してください。
0x03	SRXON	TX_ACK_DONE	TX_ACK_DONE 例外。詳細については、Table 14: Exceptions summaryを参照してください。
0x04	SSAMPLECCA	TX_UNDERFLOW	TX_UNDERFLOW 例外。詳細については、Table 14: Exceptions summaryを参照してください。
0x05	SACK	TX_OVERFLOW	TX_OVERFLOW 例外。詳細については、Table 14: Exceptions summaryを参照してください。
0x06	SACKPEND	RX_UNDERFLOW	RX_UNDERFLOW 例外。詳細については、Table 14: Exceptions summaryを参照してください。
0x07	SNACK	RX_OVERFLOW	RX_OVERFLOW 例外。詳細については、Table 14: Exceptions summaryを参照してください。
0x08	STXON	RXENABLE_ZERO	RXENABLE_ZERO 例外。詳細については、Table 14: Exceptions summaryを参照してください。
0x09	STXONCCA	RX_FRM_DONE	RX_FRM_DONE 例外。詳細については、Table 14: Exceptions summaryを参照してください。
0x0A	SFLUSHRX	RX_FRM_ACCEPTED	RX_FRM_ACCEPTED 例外。詳細については、Table 14: Exceptions summaryを参照してください。
0x0B	SFLUSHTX	SRC_MATCH_DONE	SRC_MATCH_DONE 例外。詳細については、Table 14: Exceptions summaryを参照してください。
0x0C	SRXFIFOPOP	SRC_MATCH_FOUND	SRC_MATCH_FOUND 例外。詳細については、Table 14: Exceptions summaryを参照してください。
0x0D	STXCAL	FIFOP	FIFOP 例外。詳細については、Table 14: Exceptions summaryを参照してください。
0x0E	SRFOFF	SFD	SFD 例外。詳細については、Table 14: Exceptions summaryを参照してください。
0x0F	SXOSCOFF	DPU_DONE_L	DPU_DONE_L 例外。詳細については、Table 14: Exceptions summaryを参照してください。
0x10		DPU_DONE_H	DPU_DONE_H 例外。詳細については、Table 14: Exceptions summaryを参照してください。
0x11		MEMADDR_ERROR	MEMADDR_ERROR 例外。詳細については、Table 14: Exceptions summaryを参照してください。
0x12		USAGE_ERROR	USAGE_ERROR 例外。詳細については、Table 14: Exceptions summaryを参照してください。

CTRLn (16 進)	IN(コマンド・ストロープ)	OUT	OUT 信号の説明
0x13		OPERAND_ERROR	OPERAND_ERROR 例外。詳細については、Table 14: Exceptions summaryを参照してください。
0x14		SPI_ERROR	SPI_ERROR 例外。詳細については、Table 14: Exceptions summaryを参照してください。
0x15		RF_NO_LOCK	RF_NO_LOCK 例外。詳細については、Table 14: Exceptions summaryを参照してください。
0x16		RX_FRM_ABORTED	RX_FRM_ABORTED 例外。詳細については、Table 14: Exceptions summaryを参照してください。
0x17		RXBUFMOV_TIMEOUT	RXBUFMOV_TIMEOUT 例外。詳細については、Table 14: Exceptions summaryを参照してください。
0x18		未使用	UNUSED 例外。詳細については、Table 14: Exceptions summaryを参照してください。
...		予約	
0x21		例外チャンネル A	コレクション A の 1 つまたは複数の例外フラグがアクティブな場合、ピンは "HIGH" になります。コレクション A にどの例外を含めるかを構成できます。
0x22		例外チャンネル B	コレクション B の 1 つまたは複数の例外フラグがアクティブな場合、ピンは "HIGH" になります。コレクション B にどの例外を含めるかを構成できます。
0x23		コンプリメンタリ例外チャンネル A	コレクション A がない 1 つまたは複数の例外フラグがアクティブな場合、ピンは "HIGH" になります。
0x24		コンプリメンタリ例外チャンネル B	コレクション B がない 1 つまたは複数の例外フラグがアクティブな場合、ピンは "HIGH" になります。
0x25		RX 関連エラーの事前定義例外チャンネル。	事前定義例外チャンネル。例外フラグ RX_UNDERFLOW、RX_OVERFLOW、RX_FRM_ABORTED、および RXBUFMOV_TIMEOUT のうち 1 つまたは複数の例外フラグがアクティブの場合は "HIGH" になります。
0x26		一般的なエラーの事前定義例外チャンネル。	例外フラグ MEMADDR_ERROR、USAGE_ERROR、OPERAND_ERROR、および SPI_ERROR のうち 1 つまたは複数の例外フラグがアクティブの場合は "HIGH" になります。
0x27		fifo	1 つまたは複数のバイトが RXFIFO にある場合、ピンは "HIGH" になります。RXFIFO オーバーフローが発生した場合は "LOW" になります。
0x28		fifop	RXFIFO 内のバイト数がプログラマブル・スレッシュホールドを超えるか、RXFIFO 内に完全フレームが 1 つ以上ある場合、ピンは "HIGH" になります。RXFIFO オーバーフローが発生した場合も "HIGH" になります。FIFOP 例外と混同しないでください。
0x29		cca	クリア・チャンネル・アセスメント。この信号の動作を構成する方法については、FSMSTAT1 レジスタを参照してください。

CTRLn (16 進)	IN(コマンド・ストロープ)	OUT	OUT 信号の説明
			ださい。
0x2A		sfd	SFD が受信済みまたは送信済みの場合、ピンは "HIGH" になります。RX/TX を終了するたびにクリアされます。SFD 例外と混同しないでください。
0x2B		lock	周波数シンセサイザがロックされている場合、ピンは "HIGH" になります。
0x2C		rssi_valid	RX が起動してから RSSI 値が 1 回以上更新されている場合、ピンは "HIGH" になります。RX を終了するとクリアされます。
0x2D		sampled_cca	復調器からのサンプル・バージョンの CCA ビット。SSAMPLECCA または STXONCCA ストロープが発行されるたびに値が更新されます。
0x2E		rand_i	レシーバの I チャネルからのランダム・データ出力。8MHz で更新されます。
0x2F		rand_q	レシーバの Q チャネルからのランダム・データ出力。8MHz で更新されます。
0x30		rand_xor_i_q	I と Q ランダム出力間の XOR。8MHz で更新されます。
0x31		sniff_clk	Packet Sniffer データの 250kHz クロック。
0x32		sniff_data	Packet Sniffer からのデータ。sniff_clk の立ち上がりエッジでのサンプル・データ。
0x33		mod_serial_clk	変調器からの 250kHz シリアル・データ・クロック。
0x34		mod_serial_data	変調器からのシリアル データ。mod_serial_clk の立ち上がりエッジでのサンプル・データ。
...		予約	
0x43		rx_active	FFCTRL が RX 状態の 1 つであることを示します。アクティブ "HIGH"。 注: この信号には出力フリップ・フロップがなく、FFCTRL FSM の現在の状態のレジスタをベースにしているため、グリッチが発生する可能性があります。
0x44		tx_active	FFCTRL が TX 状態の 1 つであることを示します。アクティブ "HIGH"。 注: この信号には出力フリップ・フロップがなく、FFCTRL FSM の現在の状態のレジスタをベースにしているため、グリッチが発生する可能性があります。
...		予約	
0x5E		dpu_core_activepri (0)	DPU が優先順位の低いスレッドの処理でビジーな場合は、"HIGH" になります。
0x5F		dpu_core_activepri (1)	DPU が優先順位の高いスレッドの処理でビジーな場合は、"HIGH" になります。
...		予約	

CTRLn (16進)	IN(コマンド・ストロープ)	OUT	OUT 信号の説明
0x62		dpu_state_l_active	優先順位の低いスレッドが保留中またはアクティブな場合は、"HIGH" になります。
0x63		dpu_state_h_active	優先順位の高いスレッドが保留中またはアクティブな場合は、"HIGH" になります。
...		予約	
0x7E		'0'	定数値
0x7F		'1'	定数値

12 電力モード

CC2520 には、下で説明するように 3 つの電力モードがあります。これらの電力モードのいずれの場合も、電源電圧は回路に適用されます。

低電力モード 2 (LPM2) では、デジタル電圧レギュレータはオフ (VREG_EN=0) で、実行中のクロックはありません。データは保持されません。すべてのアナログ・モジュールは、パワー・ダウン状態です。

低電力モード 1 (LPM1) では、デジタル電圧レギュレータはオン (VREG_EN=1) ですが、実行中のクロックはありません。データは保持されます。アナログ・モジュールへのパワー・ダウン信号は、デジタル部で制御されます。

アクティブ・モード では、デジタル電圧レギュレータはオン (VREG_EN=1) で、クリスタル・オシレータ・クロックが実行中です。アナログ・モジュールへのパワー・ダウン信号は、デジタル部で制御されます。

12.1 電力モードの切り替え

デバイスが LPM2 であると、レジスタ内容はすべて失われます。デバイスをアクティブ・モードにするには、リセットする必要があります。リセットしない場合、デバイスの状態は不定です。RESETn ピンを "LOW" に設定するか、SPI に対してリセット命令 (SRES) を発行することで、リセットできます。クリスタル・オシレータの制御起動および自動起動を実行できるため、RESETn 方式の使用を推奨します。

LPM2 に切り替える前にデバイスを必ずリセットしてください。これにより、デジタル部への電力供給が除去された場合に構成が常に同じになり、電流スパイクや電源が断たれたときに発生するその他の問題は起こりにくくなります。

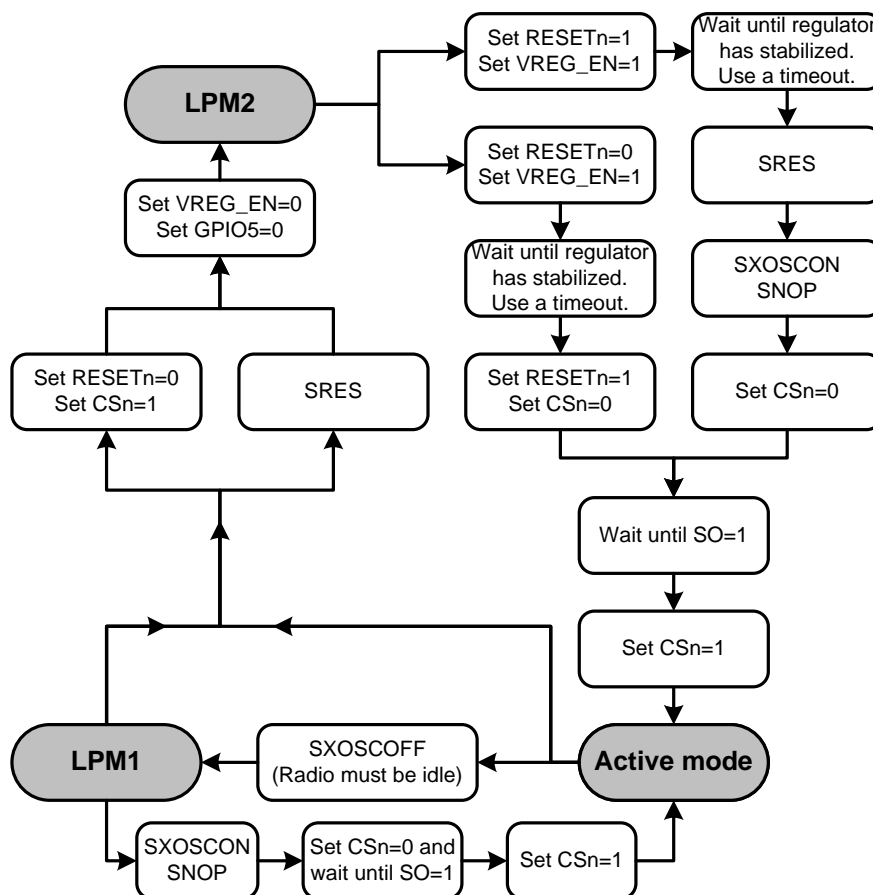


図 7: 電力モードの切り替え手順

12.2 RESETn を使用した電源投入シーケンス(推奨)

RESETn ピンを使用する場合は、内部レギュレータが安定するまでピンを "LOW" にしておく必要があります。これには通常、0.1 ミリ秒かかります。RESETn ピンを "HIGH" に設定すると、クリスタル・オシレータ(CC2520 リファレンス・デザインを参照)の開始には通常 0.2 ミリ秒かかります。クリスタル固有のパラメータについては、セクション 6 を参照してください。

チップに電力が供給され、RESETn が "LOW" に保持されている場合、GPIO ピンは表 8: GPIO リセット状態に従って構成されます。

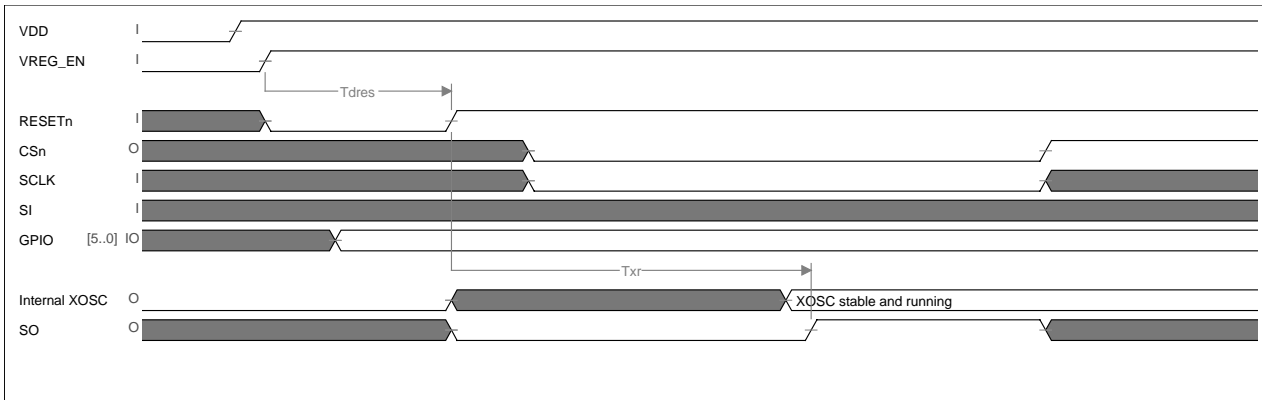


図 8: RESETn を使用した電源投入シーケンス

12.3 SRES による電源投入

電源投入後に SRES コマンド・ストローブを使用してデバイスをリセットする場合、CSn 信号を "LOW" に設定し、内部レギュレータが安定してから SRES を発行する必要があります。SRES コマンド・ストローブが発行されるまで、チップの状態は不明になります。これは、理論的に設定が送信されていることを意味することに注意してください。

電源投入から XOSC が起動するまでの時間は、SPI で使用されるクロック周波数(最大 8MHz)およびクリスタルの起動時間によって異なります。

SRES コマンド・ストローブを発行したときにクリスタル・オシレータを自動的に起動する必要はないことに注意してください。このため、水晶発振器を起動するには、SXOSCON コマンド・ストローブも発行する必要があります。RESETn ピンと異なり、SRES コマンド・ストローブを発行してもクリスタル・オシレータの状態には影響しません。そのため、クリスタル・オシレータが誤って「オフ」の状態になった場合、SRES を発行してもクリスタル・オシレータは起動しません。

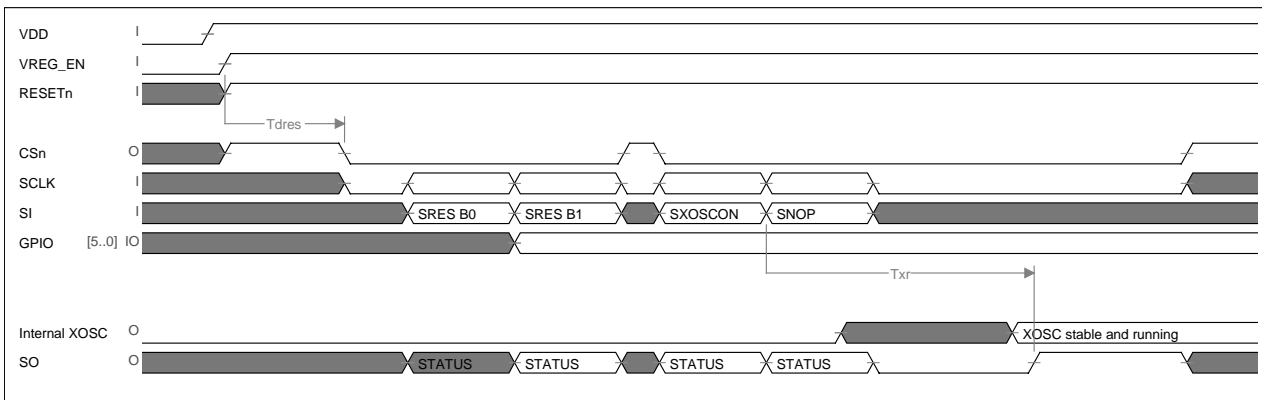


図 9: SRES を使用した電源投入シーケンス

表 10: 起動のタイミング

名称	概要	時間
Tdres	VREG_EN がアクティブになってから RESETn がリリースされるか、または CSn が "LOW" に設定されるまでに必要な時間。	≥ 0.1 ミリ秒
Txr	RESETn がリリースされるか、または SXOSCON ストローブが発行されてから内部 XOSC が安定するのにかかる時間。	0.2 ミリ秒(クリスタルによって異なる)

13 命令セット

CC2520 には総合的な命令セットがあります。命令は SPI を介して CC2520 に転送され、1 バイトまたは複数のバイトで構成されます。最初のバイトには独自の処理コードが含まれており、以下のバイトは選択した命令の実行に必要なパラメータです。次のセクションでは、各命令とパラメータの詳細について説明します。

13.1 定義

- すべてのパラメータおよびデータは、SPI を介して最上位ビットが最初で最下位ビットが最後になるよう転送されます。
- CC2520 からデータを読み取る命令の場合、データ・バイトによって SO ピンのステータス・バイトが置き換えられます。
- アドレス・パラメータは、データ・ブロック内で最下位のバイトを指します。アドレス A+1 には、最下位から次のバイトが含まれ、このパターンが繰り返されます。
- CC2520 でアドレスが自動的に増加する場合、最上位のアドレス(0xFFF)を超えると、最下位のアドレスに戻ります。
- 命令を終了するには、完了命令を送信するか(有限命令の場合)、CSn を引き上げ(命令概要の "... " で示されている無限命令の場合)ます。
- 命令が終了すると、新しい命令を開始できます。
- 命令が完了する前に終了した場合、または命令が認識されなかった場合は、OPERAND_ERROR 例外が発生します。
- ユーザが命令概要表で '0' として明示的にマークされたパラメータ・ビットを '1' に設定すると、OPERAND_ERROR 例外が発生します。
- 命令が中止されると、例外エラーが発生し、CSn を "HIGH" に設定してから再び "LOW" に設定するまで、SPI インターフェイスはデータの受信を停止します。中止された命令は、中止される前にメモリの内容を変更した可能性があります。
- SPI の(バイト間ではない)バイト転送の最中に(CSn を "HIGH" に引き上げて)SPI インターフェイスをリセットすると、SPI_ERROR 例外が発生します。

13.2 命令の説明

下に示したコードは、命令の説明に使用します。コードは、ユーザによって選択可能なビットを表します。したがって、複数のバイト全体で繰り返しの文字数と幅が同じで、このエンコーディングによる最初のバイト転送で左端のビットが MSB のワードを表している場合でも、ビットのシーケンスは同じ文字によって表されます。同じ文字を小文字でのみ表されたエンコーディングのビットでエンコードされた正の整数として解釈する必要がある場合、このようなワードは、テキスト内でエンコーディング文字の大文字で表すことができます。

1 つの整数を参照するビットは、エンコーディングで連続している必要はないことに注意してください。このため、エンコーディング `aaaaeeee aaaaaaaa eeeeeeee` は、最初のバイトで転送された各ワードの最上位ビットが含まれた 3 つのバイトで転送された 2 つの 12 ビット・ワードを表します。

表 11: 命令セットの説明で使用されるコード

コード	概要
a, e, k, n	アドレス・データ
b	ビット・アドレス
i	命令
d	データ
s	ステータス・バイト
p	優先順位
m	セキュリティ・パラメータ
c, f	カウント
-	使用しない

表 12: CC2520 の命令セット

OP コード	入力	出力	概要	例外
ペリフェラル命令				
IBUFLD	i[7:0]	s[7:0]	命令を命令バッファにロードします。命令バッファは、1 つの命令を 1 バイトの長さに保持します。ロードされる命令 I は保持され、SIBUFEX の実行時には、それらのバイトが SPI インターフェイス転送されたばかりであるかのように、通常の命令として解釈する必要があります。 命令バッファに保持された命令が実行されると、SNOP に置き換わります。	
SIBUFEX コマンド・ストローブ		s[7:0]	それらのバイトが SPI インターフェイス転送されたばかりであるかのように、命令バッファに格納されている命令を実行します。 格納された命令が命令バッファでの使用に有効でない場合、USAGE_ERROR 例外が発生します。 実行された命令によって、通常可能な例外を発生させることができます。	USAGE_ERROR スペシャル。
SSAMPLECCA コマンド・ストローブ		s[7:0]	CCA ステータス信号の値をサンプリングし、ステータス・レジスタに格納します。	
SNOP		s[7:0]	動作なし(ステータス・ビットの読み取り以外の影響はありません)	

OP コード	入力	出力	概要	例外
SXOSCON		s[7:0]	<p>クリスタル・オシレータをオンにします。XOSC がすでにオンの場合にこの命令を実行すると、命令のエフェクトはありません。</p> <p>この命令は、CSn が "LOW" に引き下げられた後の最初の命令としてのみ実行できます。</p> <p>正常に終了するためには、ただちに SNOP 命令を実行する必要があります。</p>	OPERAND_ERROR
STXCAL コマンド・ストローブ		s[7:0]	<p>TX の周波数シンセサイザを有効にしてキャリブレーションします。RX/TX からシンセサイザのみが実行中の待機状態に移動します。テスト目的のみ。</p> <p>フレームが現在受信中の場合、RX_FRM_ABORTED 例外が発生します。</p>	RX_FRM_ABORTED
SRXON コマンド・ストローブ		s[7:0]	<p>RX を有効にします。</p> <p>フレームが現在受信中の場合、RX_FRM_ABORTED 例外が発生します。</p>	RX_FRM_ABORTED
STXON コマンド・ストローブ		s[7:0]	<p>キャリブレーション後(まだ実行されていない場合)に TX を有効にします。</p> <p>フレームが現在受信中の場合、RX_FRM_ABORTED 例外が発生します。</p>	RX_FRM_ABORTED
STXONCCA コマンド・ストローブ		s[7:0]	<p>CCA がクリア・チャネルを示している場合は、次の操作を実行します。</p> <p>最初にキャリブレーションを有効にしてから TX を有効にします。</p> <p>それ以外の場合は、何もしません。</p> <p>また、CCA ステータス信号の値をサンプリングし、ステータス・レジスタに格納します。</p>	
SRFOFF コマンド・ストローブ		s[7:0]	<p>RX/TX および周波数シンセサイザ無効にします。</p> <p>RX、TX、および周波数シンセサイザがすでにオフの場合、USAGE_ERROR 例外が発生し、命令に影響はありません。</p> <p>フレームが現在受信中の場合、RX_FRM_ABORTED 例外が発生します。</p>	USAGE_ERROR RX_FRM_ABORTED
SXOSCOFF コマンド・ストローブ		s[7:0]	<p>クリスタル・オシレータをオフにします。</p> <p>RF セクションがアイドルの場合、USAGE_ERROR が生成されます。</p> <p>フレームが現在受信中の場合、RX_FRM_ABORTED 例外が発生します。</p>	USAGE_ERROR RX_FRM_ABORTED

OP コード	入力	出力	概要	例外
SFLUSHRX コマンド・ストローブ		s[7:0]	RX FIFO をフラッシュして、復調器をリセットします。 フレームが現在受信中の場合、RX_FRM_ABORTED 例外が発生します。	RX_FRM_ABORTED
SFLUSHTX コマンド・ストローブ		s[7:0]	TX FIFO をフラッシュします。	
SACK コマンド・ストローブ		s[7:0]	現在のフレームを受信後、フレームのサブフィールドのクリアを保留にしてアクノリッジ・フレームを送信します。 フレームが現在受信中でない場合は、USAGE_ERROR 例外が発生させます。この場合、ACK フレームは送信されません。	USAGE_ERROR
SACKPEND コマンド・ストローブ		s[7:0]	現在のフレームを受信後、フレームのサブフィールドの設定を保留にしてアクノリッジ・フレームを送信します。 フレームが現在受信中でない場合は、USAGE_ERROR 例外が発生させます。この場合、ACK フレームは送信されません。	USAGE_ERROR
SNACK コマンド・ストローブ		s[7:0]	rfr_autoack が設定されている場合でも、現在受信済みのフレームにはアクノリッジ・フレームを送信しないでください。 フレームが現在受信中でない場合は、USAGE_ERROR 例外が発生させます。この場合、ACK フレームは送信されません。	USAGE_ERROR
SRXMASKBITSET コマンド・ストローブ		s[7:0]	RXMASK のビット 13 を設定します。	
SRXMASKBITCLR コマンド・ストローブ		s[7:0]	RXMASK のビット 13 をクリアします。 これにより、RXENABE レジスタの値がゼロになった場合、RXENABLE_ZERO 例外が発生させます。	RXENABLE_ZERO
SRXMASKOR	d[15:0]	s[7:0]	RX イネーブル・マスクと D 間でビット単位の OR 演算を実行します。	
SRXMASKAND	d[15:0]	s[7:0]	RX イネーブル・マスクと D 間でビット単位の AND 演算を実行します。これにより、RXENABLE レジスタの値がゼロになった場合、RXENABLE_ZERO 例外が発生させます。	RXENABLE_ZERO
データ IO				
BSET	a[7:3] b[2:0]	s[7:0]	シングル・ビットを設定します。アドレス A のビット B に 1 を書き込みます。これを実行しても、アドレスの値は変わらず、同じアドレスのその他のビットへの副作用も生じません。この命令に使用できるのは、アドレス範囲 [0, 31] のみです。	MEMADDR_ERROR

OP コード	入力	出力	概要	例外
BCLR	a[7:3] b[2:0]	s[7:0]	シングル・ビットをクリアします。アドレス A のビット B に 0 を書き込みます。これを実行しても、アドレスの値は変わらず、同じアドレスのその他のビットへの副作用も生じません。この命令に使用できるのは、アドレス範囲 [0, 31] のみです。	MEMADDR_ERROR
MEMRD	a[11:0]	s[7:0] d[7:0] ...	メモリを読み取ります。データ D の n 番目のバイトがアドレス(A+n)から読み取られます。LSB=0 のアドレスを読み取ると、対応する LSB=1 のアドレスの内容はバッファに格納されることに注意してください。同じ MEMRD 命令の直後にこのアドレスを読み取ると、バッファに格納されたアドレスのコピーが読み取られます。この方法で、16 ビット・ワード全体の読み取りは、アトミック処理として実行されます。	MEMADDR_ERROR
MEMWR	a[11:0] d[7:0] ...	s[7:0] d[7:0] ...	メモリに書き込みます。命令によって入力されたデータ D の n 番目のバイトがアドレス(A+n)に書き込まれます。 さらに、命令によって出力されたデータ D の n 番目のバイトは、メモリ位置(A+n)から読み取られた未修正のデータです。	MEMADDR_ERROR
REGRD	a[5:0]	s[7:0] d[7:0] ...	MEMRD と同じ機能。ただし、0x40 より下のアドレスでのみ処理を開始できます。	MEMADDR_ERROR
REGWR	a[5:0] d[7:0] ...	s[7:0] d[7:0] ...	MEMWR と同じ機能。ただし、0x40 より下のアドレスでのみ処理を開始できます。	MEMADDR_ERROR
MEMXWR	a[11:0] d[7:0] ...	s[7:0] d[7:0] ...	XOR メモリ。指示に従って、n 番目のデータ・バイト D のビット単位の XOR 演算を書き込み、アドレス(A+n)の現在の内容をメモリ位置(A+n)に書き込みます。 さらに、命令によって出力されたデータ D の n 番目のバイトは、メモリ位置(A+n)から読み取られた未修正のデータです。	MEMADDR_ERROR
RXBUF		s[7:0] d[7:0] ...	RX FIFO の最も古いバイトを読み取ります。最初のデータ転送で、RX FIFO の最も古いバイトが読み取られ、RX FIFO から削除されます。この処理は、後続の SPI 転送で繰り返されます。 RX FIFO が空の場合、この命令を実行すると、RX_UNDERFLOW 例外が発生します。 注: RXBUFMOV が進行中の場合は、RXBUF を実行しないでください。実行すると、データが失われます。	RX_UNDERFLOW

OP コード	入力	出力	概要	例外
RXBUFCP	a[11:0]	s[7:0] c[7:0] d[7:0] ...	この命令は RXBUF として機能しますが、RX FIFO から読み取られたデータ・バイトをアドレス A で始まるメモリ位置にコピーします。 転送された 2 番目のバイトは、RX FIFO 内の現在のバイト数 C を表します。 注: RXBUFMOV が進行中の場合は、RXBUFCP を実行しないでください。実行すると、データが失われます。	RX_UNDERFLOW
TXBUF	d[7:0] ...	s[7:0] c[7:0]	TX FIFO の最後に書き込みます。OP コードの後に転送されたデータ・バイトは、TX FIFO の最後に追加されます。 SPI インターフェイスは、現在転送済みのバイトを入力する前に、TX FIFO 内のバイト数 C を出力します。最初のバイトを TX FIFO に転送すると、0x00 が返されます。 TX FIFO がフルの場合、この命令を実行すると、TX_OVERFLOW 例外が発生します。	TX_OVERFLOW
RANDOM	...	s[7:0] d[7:0] ...	受信チェーンのノイズから生成されたバイト D をランダムに読み取ります。	
データ管理命令				
RXBUFMOV	P a[11:0] c[7:0]	s[7:0] c[7:0]	C の最も古いバイトを RX FIFO からアドレス A で始まるメモリ位置に移動します。 命令の優先順位は P で表されます。P=0 は優先順位が低いことを表し、P=1 は優先順位が高いことを表します。 命令が完了する前に RX FIFO が空になった場合は、DPUCON.RXTIM で定義されているように RXBUFMOV_TIMEOUT 例外が発生します。移動する残りのバイトは、ステータス・レジスタ内で使用できます。DPUCON.RXTIM='1' に設定して高い優先順位で RXBUFMOV を実行すると、フレームの受信(128 バイト・フレームの場合は 4 ミリ秒以上)は、その他の DPU 命令の実行がブロックされることに注意してください。 処理が完了すると、命令の優先順位に応じて DPU_DONE_L または DPU_DONE_H 例外が発生します。この例外は、処理が正常に完了したかどうかに関係なく発生します。 優先順位レベル(高または低)を指定した命令がすでにアクティブの場合、USAGE_ERROR 例外が発生します。	RXBUFMOV_TIMEOUT OPERAND_ERROR USAGE_ERROR DPU_DONE_L DPU_DONE_H MEMADDR_ERROR

OP コード	入力	出力	概要	例外
TXBUFCP	P a[11:0] c[7:0]	s[7:0] c[7:0]	<p>アドレス AT T から開始するメモリ位置から C バイトのデータを TXBUF の最後にコピーします。</p> <p>SPI インターフェイスは、TXBUF のバイト数 C を出力します。</p> <p>命令の優先順位は P で表されます。P=0 は優先順位が低いことを表し、P=1 は優先順位が高いことを表します。</p> <p>処理が完了する前に TXBUF がフルになると、TX_OVERFLOW 例外が発生します。移動する残りのバイトは、ステータスレジスタ内で使用できません。</p> <p>処理が完了すると、命令の優先順位に応じて DPU_DONE_L または DPU_DONE_H 例外が発生します。この例外は、処理が正常に完了したかどうかに関係なく発生します。</p> <p>優先順位レベル(高または低)を指定した命令がすでにアクティブの場合、USAGE_ERROR 例外が発生します。</p>	<p>TX_OVERFLOW OPERAND_ERROR USAGE_ERROR DPU_DONE_L DPU_DONE_H MEMADDR_ERROR</p>
MEMCP	P c[7:0] a[11:0] e[11:0]	s[7:0]	<p>1 つのメモリブロックから別のメモリブロックにデータをコピーします。アドレス A で始まるメモリ位置から C バイトのデータのブロックをアドレス E で始まるメモリ位置にコピーします。</p> <p>命令の優先順位は P で表されます。P=0 は優先順位が低いことを表し、P=1 は優先順位が高いことを表します。</p> <p>処理が完了すると、命令の優先順位に応じて DPU_DONE_L または DPU_DONE_H 例外が発生します。この例外は、処理が正常に完了したかどうかに関係なく発生します。</p> <p>優先順位レベル(高または低)を指定した命令がすでにアクティブの場合、USAGE_ERROR 例外が発生します。</p>	<p>MEMADDR_ERROR USAGE_ERROR DPU_DONE_L DPU_DONE_H</p>

OP コード	入力	出力	概要	例外
MEMCPR	P c[7:0] a[11:0] e[11:0]	s[7:0]	<p>1 つのメモリ・ブロックから別のメモリ・ブロックにデータをコピーし、バイトの並び順を反転します。アドレス A で始まるメモリ位置から C バイトのデータのブロックをアドレス E で始まるメモリ位置にコピーし、データ・ブロックのバイトの並び順を反転します。メモリ位置(A+n)のデータは、メモリ位置(E+C-1-n)に書き込まれます。</p> <p>命令の優先順位は P で表されます。P=0 は優先順位が低いことを表し、P=1 は優先順位が高いことを表します。</p> <p>処理が完了すると、命令の優先順位に応じて DPU_DONE_L または DPU_DONE_H 例外が発生します。この例外は、処理が正常に完了したかどうかに関係なく発生します。</p> <p>優先順位レベル(高または低)を指定した命令がすでにアクティブの場合、USAGE_ERROR 例外が発生します。</p>	MEMADDR_ERROR USAGE_ERROR DPU_DONE_L DPU_DONE_H
MEMXCP	P c[7:0] a[11:0] e[11:0]	s[7:0]	<p>1 つのメモリ・ブロックと別のメモリ・ブロックの XOR。命令への入力となる 2 つのメモリ・ブロックは、いずれのサイズも C バイトで、アドレス A および E からそれぞれ始まります。出力されるのは、2 つのメモリ・ブロックのビット単位の XOR 演算で、アドレス E で始まるメモリ位置に書き込まれます。</p> <p>命令の優先順位は P で表されます。P=0 は優先順位が低いことを表し、P=1 は優先順位が高いことを表します。</p> <p>処理が完了すると、命令の優先順位に応じて DPU_DONE_L または DPU_DONE_H 例外が発生します。この例外は、処理が正常に完了したかどうかに関係なく発生します。</p> <p>優先順位レベル(高または低)を指定した命令がすでにアクティブの場合、USAGE_ERROR 例外が発生します。</p>	MEMADDR_ERROR DPU_DONE_L DPU_DONE_H USAGE_ERROR

OP コード	入力	出力	概要	例外
セキュリティ命令				
INC	P c[1:0] a[11:0]	s[7:0]	<p>2^C バイト・ワードをアドレス A の最下位バイトでインクリメントします。最下位バイトから n 番目のバイトはアドレス (A+n) にあります (n' < n は n' が n より下位にあることを意味します)。</p> <p>C は 0~2 の範囲内で、1、2、または 4 バイトがインクリメントされます。C が 3 の場合、USAGE_ERROR 例外が発生します。</p> <p>命令の優先順位は P で表されます。P=0 は優先順位が低いことを表し、P=1 は優先順位が高いことを表します。</p> <p>優先順位レベル(高または低)を指定した命令がすでにアクティブの場合、USAGE_ERROR 例外が発生します。</p> <p>処理が完了すると、命令の優先順位に応じて DPU_DONE_L または DPU_DONE_H 例外が発生します。この例外は、処理が正常に完了したかどうかに関係なく発生します。</p> <p>C パラメータの設定に関係なく、命令は常に 4 バイトにアクセスするため、A > 0x3FC の場合は MEMADDR_ERROR 例外が発生します。</p>	MEMADDR_ERROR USAGE_ERROR DPU_DONE_L DPU_DONE_H
ECB	P k[7:0] c[3:0] a[11:0] e[11:0]	s[7:0]	<p>ECB 暗号化。メモリから読み取った(16-C)バイトのデータで構成され、C ゼロバイトで連結された、アドレス A で始まるテキスト形式の 1 つの 16 バイト・ブロックをアドレスに格納されたキー(16?K)を使用して、出力をアドレス E に格納して、暗号化します。</p> <p>命令の優先順位は P で表されます。P=0 は優先順位が低いことを表し、P=1 は優先順位が高いことを表します。</p> <p>出力は 16 AES-128 で暗号化されたバイトです。</p> <p>優先順位レベル(高または低)を指定した命令がすでにアクティブの場合、USAGE_ERROR 例外が発生します。</p> <p>処理が完了すると、命令の優先順位に応じて DPU_DONE_L または DPU_DONE_H 例外が発生します。この例外は、処理が正常に完了したかどうかに関係なく発生します。</p>	MEMADDR_ERROR USAGE_ERROR DPU_DONE_L DPU_DONE_H

OP コード	入力	出力	概要	例外
ECBO	P k[7:0] c[3:0] a[11:0]	s[7:0]	<p>ECB 暗号化。メモリから読み取った(16-C)バイトのデータで構成され、C ゼロバイトで連結された、アドレス A で始まるテキストの 1 つの 16 バイト・ブロックを、アドレスに格納されたキー(16?K)を使用して、アドレス A のデータを上書きすることで出力を格納して、暗号化します。</p> <p>命令の優先順位は P で表されます。P=0 は優先順位が低いことを表し、P=1 は優先順位が高いことを表します。</p> <p>出力は 16 AES-128 で暗号化されたバイトです。</p> <p>優先順位レベル(高または低)を指定した命令がすでにアクティブの場合、USAGE_ERROR 例外が発生します。</p> <p>処理が完了すると、命令の優先順位に応じて DPU_DONE_L または DPU_DONE_H 例外が発生します。この例外は、処理が正常に完了したかどうかに関係なく発生します。</p>	MEMADDR_ERROR USAGE_ERROR DPU_DONE_L DPU_DONE_H
ECBX	P k[7:0] c[3:0] a[11:0] e[11:0]	s[7:0]	<p>ECB 暗号化と XOR。ECB は各暗号文を除外するため、バイトはアドレスに書き込む前の送信先アドレス E の既存のバイトが含まれたビット単位の XOR 演算になります。</p>	USAGE_ERROR MEMADDR_ERROR DPU_DONE_L DPU_DONE_H
CTR	P k[7:0] c[6:0] n[7:0] a[11:0] e[11:0]	s[7:0]	<p>カウンタ・モード暗号化を使用した暗号化命令。アドレスに格納されたキー(16?K)、アドレスに格納されたカウンタ(16?N)を使用して、アドレス A で始まる C バイトのテキストを処理し、アドレス E で始まる出力を格納します。</p> <p>命令の優先順位は P で表されます。P=0 は優先順位が低いことを表し、P=1 は優先順位が高いことを表します。</p> <p>命令で指定された送信先アドレス E がゼロの場合、送信先アドレス E は A に設定され、テキストは暗号文で直接置き換えられます(UCTR 命令の場合は、暗号文がテキストで置き換えられます)。</p> <p>出力は、セキュリティレベル 4(CTR)の 802.15.4 標準に従って処理された C 暗号化バイトです。</p> <p>優先順位レベル(高または低)を指定した命令がすでにアクティブの場合、USAGE_ERROR 例外が発生します。</p> <p>処理が完了すると、命令の優先順位に応じて DPU_DONE_L または DPU_DONE_H 例外が発生します。この例外は、処理が正常に完了したかどうかに関係なく発生します。</p>	MEMADDR_ERROR USAGE_ERROR DPU_DONE_L DPU_DONE_H

OP コード	入力	出力	概要	例外
UCTR	P k[7:0] c[6:0] n[7:0] a[11:0] e[11:0]	s[7:0]	CTR モード復号化を使用した復号化命令。 カウンタ・モードの暗号化と復号化は対照的な処理であるため、この命令は CTR と同じです。	MEMADDR_ERROR USAGE_ERROR DPU_DONE_L DPU_DONE_H
CBCMAC	P k[7:0] c[6:0] a[11:0] e[11:0] m[2:0]	s[7:0]	CBC-MAC セキュリティを使用した認証命令。 アドレスに格納されたキー(16?K)T T を使用してアドレス A で始まる C バイトのテキストを処理し、アドレス E で始まる出力を格納します。 命令の優先順位は P で表されます。P=0 は優先順位が低いことを表し、P=1 は優先順位が高いことを表します。 命令で指定された送信先アドレス E がゼロの場合、送信先アドレス E は(A + C)に設定され、テキスト入力データに続いて出力が直接書き込まれます。 出力は、命令 M[1:0] が 1、2、または 3 の場合に対応する 4、8、または 16 バイトの完全性符号です。M[1:0]=0 の場合、完全性符号は出力されません。 M[2]=0 の場合、認証されるテキスト・データには、IEEE 802.15.4-2003 で使用される C というプリフィックスが自動的に付きます。 M[2]=1 の場合、テキスト・データには C というプリフィックスは付きません。このモードは、既存のシステムとの後方互換性を保つために使用できます。 優先順位レベル(高または低)を指定した命令がすでにアクティブの場合、USAGE_ERROR 例外が発生します。 処理が完了すると、命令の優先順位に応じて DPU_DONE_L または DPU_DONE_H 例外が発生します。この例外は、処理が正常に完了したかどうかに関係なく発生します。	MEMADDR_ERROR USAGE_ERROR DPU_DONE_L DPU_DONE_H OPERAND_ERROR

OP コード	入力	出力	概要	例外
UCBCMAC	P k[7:0] c[6:0] a[11:0] m[2:0]	s[7:0]	<p>CBC-MAC セキュリティを使用した逆認証命令。アドレスに格納されたキー(16?K)を使用して、アドレス A で始まる C バイトのテキストを処理します。</p> <p>命令の優先順位は P で表されます。P=0 は優先順位が低いことを表し、P=1 は優先順位が高いことを表します。</p> <p>命令によって 4、8、または 16 バイトの完全性符号が生成され(M[1:0] が 1、2、または 3 の場合)、アドレス(A + C)で受信した完全性符号と比較されます。結果(合格または不合格)は、セキュリティ処理の優先順位の高低に応じてそれぞれ AUTHSH または AUTHSL ステータス・ビットに格納されます。M[1:0]=0 の場合、完全性符号のチェックは実行されず、結果は常に「合格」になります。</p> <p>M[2]=0 の場合、認証されるテキスト・データには、IEEE 802.15.4-2003 で使用される C というプリフィックスが自動的に付きます。</p> <p>M[2]=1 の場合、テキスト・データには C というプリフィックスは付きません。このモードは、既存のシステムとの後方互換性を保つために使用できます。</p> <p>優先順位レベル(高または低)を指定した命令がすでにアクティブの場合、USAGE_ERROR 例外が発生します。</p> <p>処理が完了すると、命令の優先順位に応じて DPU_DONE_L または DPU_DONE_H 例外が発生します。この例外は、処理が正常に完了したかどうかに関係なく発生します。</p>	MEMADDR_ERROR USAGE_ERROR DPU_DONE_L DPU_DONE_H

OP コード	入力	出力	概要	例外
CCM	P k[7:0] c[6:0] n[7:0] a[11:0] e[11:0] f[6:0] m[1:0]	s[7:0]	<p>CCM / CCM* セキュリティを使用した暗号化および認証命令。アドレス A で始まる F バイトのテキストを認証します。アドレス(A+F)で始まる C バイトを認証および暗号化します。アドレスに格納されたキー(16?K)、アドレスで始まるカウンタ(16?N)を使用して、アドレス E で始まる出力を格納します。</p> <p>命令の優先順位は P で表されます。P=0 は優先順位が低いことを表し、P=1 は優先順位が高いことを表します。</p> <p>命令で指定された送信先アドレス E がゼロの場合、送信先アドレス E は(A+F)に設定され、テキストの最後の C バイトは暗号文および完全性符号に置き換わります。</p> <p>出力は C 暗号化バイトと、それに続く M が 0、1、2、または 3 の場合に対応する 0、4、8、または 16 バイトの完全性符号です。</p> <p>優先順位レベル(高または低)を指定した命令がすでにアクティブの場合、USAGE_ERROR 例外が発生します。</p> <p>((C+F) > 128)の場合は、USAGE_ERROR 例外も発生します。</p> <p>処理が完了すると、命令の優先順位に応じて DPU_DONE_L または DPU_DONE_H 例外が発生します。この例外は、処理が正常に完了したかどうかに関係なく発生します。</p>	MEMADDR_ERROR USAGE_ERROR DPU_DONE_L DPU_DONE_H

OP コード	入力	出力	概要	例外
UCCM	P k[7:0] c[6:0] n[7:0] a[11:0] e[11:0] f[6:0] m[1:0]	s[7:0]	<p>CCM / CCM* セキュリティを使用した復号化と逆認証命令。アドレス A で始まる F バイトのテキストを認証します。アドレス(A+F)で始まる C バイトを復号化と認証を行います。アドレスに格納されたキー(16?K)T、T アドレスに格納されたカウンタ(16?N)を使用して、アドレス E で始まる出力を格納します。</p> <p>命令の優先順位は P で表されます。P=0 は優先順位が低いことを表し、P=1 は優先順位が高いことを表します。</p> <p>出力は C テキスト・バイトです。引き継ぐ命令の認証部分では、これらのバイトを暗号文の読み取り元(A+F)のアドレスに書き戻す必要があることに注意してください。これを行うには、E=0x000 に設定します。</p> <p>また、命令は 0、4、8、または 16 バイトの暗号化された完全性符号(M が 0、1、2、または 3 の場合)を生成し、アドレス(A+C+F)に格納された完全性符号と比較します。結果(合格または不合格)は、セキュリティ処理の優先順位の高低に応じてそれぞれ AUTHSH または AUTHSL ステータス・ビットに格納されます。</p> <p>優先順位レベル(高または低)を指定した命令がすでにアクティブの場合、USAGE_ERROR 例外が発生します。</p> <p>((C+F) > 128)の場合は、USAGE_ERROR 例外も発生します。</p> <p>処理が完了すると、命令の優先順位に応じて DPU_DONE_L または DPU_DONE_H 例外が発生します。この例外は、処理が正常に完了したかどうかに関係なく発生します。</p>	MEMADDR_ERROR USAGE_ERROR DPU_DONE_L DPU_DONE_H
その他				
ABORT	c[1:0]	s[7:0]	<p>継続的なデータ管理またはセキュリティ命令を中止します。</p> <p>c[1]=1: 優先順位の高いデータ管理またはセキュリティ命令を中止します。 c[0]=1: 優先順位の低いデータ管理またはセキュリティ命令を中止します。 c[1]=0: 優先順位の高いデータ管理またはセキュリティ命令を中止しません。 c[0]=0: 優先順位の低いデータ管理またはセキュリティ命令を中止しません。</p> <p>命令のクラスが中止されると、継続的な命令はただちに終了し、デバイス状態はその時のままになります。保留中のすべてのデータ管理命令はフラッシュされます。</p>	

OP コード	入力	出力	概要	例外
SRES		s[7:0]	SPI インターフェイス以外のデバイスをリセットします。 この命令は、CSn が "LOW" に引き下げられた後の最初の命令としてのみ実行できます。	

13.3 命令セットの概要

CC2520 命令セットの概要と OP コードを下の表に示します。

13.4 ステータス・バイト

SPI を介して CC2520 に送信されたすべての命令により、命令の最初のバイトが SI でクロック入力されると、ステータス・バイトが SO に出力されます。立ち下がリエッジが CSn および各バイト内の SCLK の最後の立ち下がリエッジで検出されると、ステータス・バイトは内部でラッチされます。ラッチ・ステータス値は、以下の SCLK 立ち下がリエッジにシフトします。

SNOP 命令は、副作用を発生させずにステータス・バイトを読み取るのに使用できます。

表 13: ステータス・バイトの内容

ステータス・バイト(最初にシリアル出力された MSB)		
ビット番号	信号	概要
7	XOSC が安定している	0: XOSC はオフで、まだ安定していない 1: XOSC は安定していて実行中(デジタル部にクロックあり)
6	RSSI が有効	0: RSSI 値は有効でない 1: RSSI 値は有効
5	EXCEPTION チャンネル A	0: EXCMASKAn で選択した例外には、EXCFLAGn セットの対応するフラグがない 1: EXCMASKAn で選択した少なくとも 1 つの例外に、対応するフラグの EXCFLAGn セットが含まれている
4	EXCEPTION チャンネル B	0: EXCMASKBn で選択した例外には、EXCFLAGn セットの対応するフラグがない 1: EXCMASKBn で選択した少なくとも 1 つの例外に、対応するフラグの EXCFLAGn セットが含まれている
3	DPU H がアクティブ	0: 優先順位の高い DPU 命令で現在アクティブなものはない 1: 優先順位の高い DPU 命令は現在アクティブ
2	DPU L がアクティブ	0: 優先順位の低い DPU 命令で現在アクティブなものはない 1: 優先順位の低い DPU 命令は現在アクティブ
1	TX がアクティブ	0: デバイスは TX モードではない 1: デバイスは TX モード
0	RX がアクティブ	0: デバイスは RX モードではない 1: デバイスは RX モード

13.5 コマンド・ストローブ

セクション 15.3 の長さが 1 バイトの命令の多くは、コマンド・ストローブと呼ばれます。ただし、SNOP および SXOSCON の 2 つの命令を除きます。SNOP は、副作用を発生させずにステータス・バイトを読み取るのに使用されます。SXOSCON はクリスタル・オシレータをオンにし、SPI を介して実行する必要があります。IBUFLD を使用して SXOSCON を命令バッファにロードし、IBUFEX を使用して実行することはできません。

コマンド・ストローブは、セクション 12.6 の GPIO 構成表にしたがって GPIO ピンを入力として構成することで実行でき、GPIOPOLARITY レジスタで選択したエッジを使用してトリガできます。SPI トラフィックは、コマンド・ストローブで省略できます。

コマンド・ストローブで例外を発生させるにも X と Y という 2 つのチャネルがあるため、CC2520 は各内部イベントに自動的に反応します。この機能の詳細については、セクション 16.1 を参照してください。

13.6 コマンド・ストローブ・バッファ

コマンド・ストローブ・バッファは、コマンド・ストローブを実行する別のメカニズムです。このバッファは、SPI を介して送信された IBUFLD 命令によってロードされます。バッファにロード後、CC2520 が SIBUFEX ストローブを受信すると、命令が実行されます。SIBUFEX ストローブは、トリガ・ソース(SPI、GPIO、SIBUFEX 命令にバインドされた例外)からトリガできます。命令バッファ内の命令が実行されると、SNOP 命令に置き換わります。SIBUFEX ストローブと IBUFLD 命令の両方を同時に受信すると、古いコマンド・ストローブが実行されます。ユーザがバッファに書き込もうとした新しいストローブは失われ、二度と実行されません。

14 例外

CC2520 の例外は、さまざまなイベントの発生を示すのに使用されます。例外は、SPI の不正な使用などのエラー状態およびフレーム開始部(SFD)の送信などの完全に正常で予期されたイベントの両方に使用されます。例外フラグはステータス・レジスタに格納され、SPI を介した読み取りや GPIO での監視ができます。例外フラグをクリアするには、ステータス・レジスタの該当するビットに '0' を書き込む必要があります。例外が発生したのとまったく同じクロック期間で例外フラグをクリアしようとすると、フラグはクリアされません。

表 14 は、CC2520 で使用できる例外の概要を示したものです。番号列は、例外番号の設定方法を表しています。番号は EXCFLAGn レジスタのビットに対応しており、例外を命令にバインドする場合に使用する必要があります。

表 14: 例外の概要

ニ一モニック	番号(16 進)	概要
RF_IDLE	0x00	メインの無線 FSM がアイドル状態になります。デバイスのリセットにより、FSM がアイドル状態になった場合、この例外は生成されません。
TX_FRM_DONE	0x01	TX フレームが正常に送信されました。これは、TX FIFO が空になり、アンダーフローが発生しないことを表します。SRFOFF、SRXON、または STXON によって TX が中止された場合、例外は生成されません。
TX_ACK_DONE	0x02	ACK フレームが正常に送信されました。SRFOFF、SRXON、または STXON によってアクノリッジ送信が中止された場合、例外は生成されません。
TX_UNDERFLOW	0x03	アンダーフローが TX FIFO で発生しました。TX は中止され、TX FIFO をフラッシュする必要があります。
TX_OVERFLOW	0x04	TX FIFO がフルの状態、TX FIFO に書き込もうとしました。命令は中止されます。
RX_UNDERFLOW	0x05	読み取り可能なバイトがない状態で、RX FIFO を読み取ろうとしました。命令は中止されます。 RX_UNDERFLOW 例外は、ソフトウェアのデバッグにのみ使用でき、RX FIFO 読み出しルーチンでは信頼できないことに注意してください。状況によっては、RX_FIFO が空であっても、読み取りの開始時に RX_UNDERFLOW 例外は発行されません。
RX_OVERFLOW	0x06	RX FIFO がフルの状態、RF_core は RX FIFO に書き込もうとしました。RX FIFO に書き込もうとしたバイトは失われます。データの受信は中止され、FSM は rx_overflow 状態になります。推奨されるアクションは、SFLUSHRX コマンド・ストローブを発行し、RX FIFO を空にして、RX を再起動することです。
RXENABLE_ZERO	0x07	RX イネーブル・レジスタの値はすべてゼロに変更されました。
RX_FRM_DONE	0x08	完了フレームが受信されました。長さフィールドに設定されたバイト数が受信されます。
RX_FRM_ACCEPTED	0x09	フレーム・フィルタリングが有効な場合、フレームを受信すると、この例外が生成されます(結果の特定に必要なフィールドを受信した直後に発生します)。

ニーモニック	番号(16進)	概要
SRC_MATCH_DONE	0x0A	送信元アドレス・マッチングが有効な場合、送信元アドレス・マッチングが完了すると、この例外が生成されます。例外は、結果に関係なく生成されます。
SRC_MATCH_FOUND	0x0B	送信元アドレスの一致が見つかった場合、SRC_MATCH_DONEの直前にこの例外が生成されます。
FIFOP	0x0C	RX FIFO は、アドレス・フィルタをレジスタに定義されているFIFOP スレッシュホールド値に渡したバイトで埋められるか、または少なくとも1つの完了フレームがRX FIFOに書き込まれています。FFCTRLがx_overflow状態の場合は"HIGH"です。
SFD	0x0D	RX内の場合はフレーム開始部が受信され、TXフレーム開始部が送信されます。
DPU_DONE_L	0x0E	優先順位の低いDPU処理は完了しました。処理が失敗または中止されても、例外は発行されません。
DPU_DONE_H	0x0F	優先順位の高いDPU処理は完了しました。処理が失敗または中止されても、例外は発行されません。
MEMADDR_ERROR	0x10	不正なアドレスが命令に使用されました。命令は中止されます。
USAGE_ERROR	0x11	この命令を許可しないコンテキストで実行された命令。命令は中止されます。
OPERAND_ERROR	0x12	命令の形式が間違っています。命令は中止されます。これは、バイトの境界でCSnが引き上げられたが、必要な数のオペランドが転送される前に起こった場合、マルチバイトの固定長命令で発生します。
SPI_ERROR	0x13	CSnをバイトの中央に引き上げたことで、SPI転送は中止されました(バイト境界ではない)。
RF_NO_LOCK	0x14	RXの入力後、256usの前にロックが見つからないと、この例外がアクティブになります。また、RX内の場合、LOCK_STATUSのネガティブ・エッジによってこの例外がトリガされます。
RX_FRM_ABORTED	0x15	フレーム受信が中止されました。RX_OVERFLOWが発生しても、発行されません。
RXBUFMOV_TIMEOUT	0x16	RXBUFMOVがタイムアウトになりました。RX FIFOには使用できる十分なバイトがなく、DPUCON.RXTIMEOUTによって設定された待機時間の期限が切れました。
UNUSED	0x17	予約済み

14.1 GPIOピンでの例外

Table 8の目的の例外に対応するCTRLn値をGPIOCTRLnレジスタに書き込むことで、すべての例外フラグをGPIOピンに個別に送信できます。

CC2520にはAとBという2つの例外チャンネルがあるので、ユーザは結合する例外のコレクションを選択して、GPIOピンに出力できます。選択した例外が1つでもアクティブになると、GPIOピンもアクティブになります。また、2つの各チャンネルの例外のコンプリメンタリ・コレクションを出力することもできます。

例: 例外チャンネルBでRF_IDLEおよびRX_UNDERFLOWを収集し、GPIO3に出力します。

- Write 0x22 を GPIOCTRL3 に書き込みます。GPIO3 を出力として設定し、セクション 12.6 の GPIO 構成表から例外チャンネル B を選択します。
- 0x21 を EXCMASKB0 に書き込みます。表「例外の概要」(セクション 16)に従って、RF_IDLE および RX_UNDERFLOW 例外を選択します。
- 0x00 を EXCMASKB1 に書き込みます。その他のすべての例外をマスクします。
- 0x00 を EXCMASKB2 に書き込みます。その他のすべての例外をマスクします。

上の例の設定が含まれたコンプリメンタリ例外チャンネル B には、RF_IDLE と RX_UNDERFLOW 以外のすべての例外が含まれます。対応する GPIOCTRLn レジスタに 0x24 を書き込むことで、このチャンネルを別の GPIO ピンに送信できます。

個別またはチャンネル内のグループとして GPIO ピンに関連付けられた例外は、EXCFLAGn レジスタの対応するビットに一致します。ビットにゼロを書き込むことでクリアできるのは、"HIGH" に設定された EXCFLAGn レジスタ・ビットのみです。

14.2 事前定義例外チャンネル

GPIO ピンで監視できる事前定義例外チャンネルは 2 つあります。これらのチャンネルはステータス・バイトには含まれず、使用可能なコンプリメンタリ・チャンネルもありません。

最初の事前定義例外チャンネルは、RX の実行中に問題が発生したことを示す例外の集まりです。

2 番目の事前定義例外チャンネルには、一般的なエラー状態を示す例外が含まれています。

図 10 は、例外を CC2520 の命令セットに関連付ける方法を示しています。命令をトリガできるソースは複数あることに注意してください。大きな OR ゲートは、命令の実行をトリガするソースがいずれか 1 つであることを表しています。

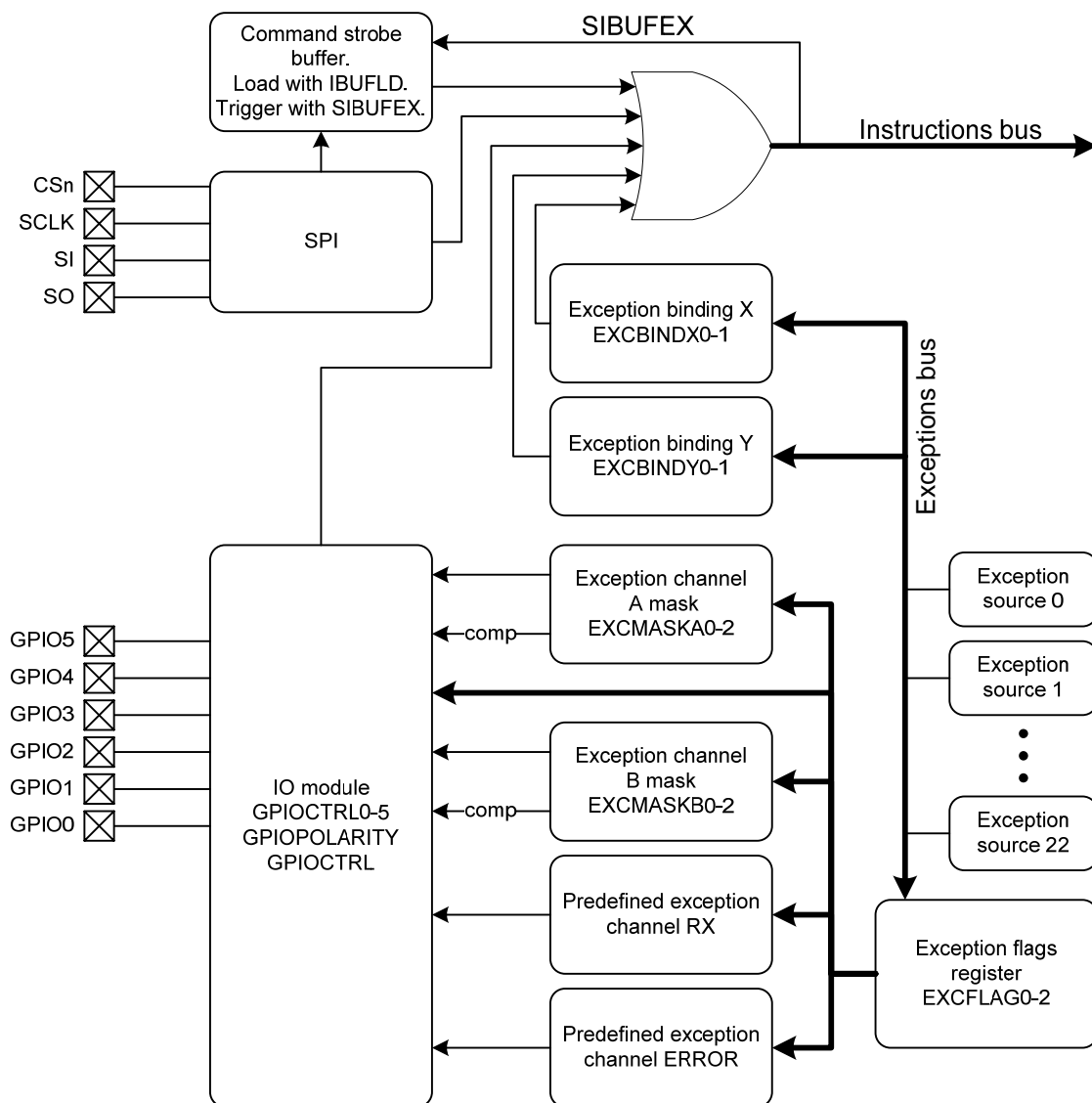


図 10: 例外処理および命令のトリガの機能詳細

14.3 例外と命令のバインド(コマンド・ストローブ)

例外が発生した場合にコマンド・ストローブを自動的に実行するよう、例外をバインドしてコマンド・ストローブをトリガできます。バインドの組み合わせには X と Y の 2 種類があり、レジスタ EXCBINDXn および EXCBINDYn で定義されています。

例

RX_FRM_ACCEPTED 例外がアクティブの場合、SACKPEND 命令を実行します。

1. 0x06 を EXCBINDX0 に書き込みます。これにより、バインドされた命令として SACKPEND が Table 8: GPIO configuration から選択されます。
2. 0x89 を EXCBINDX1 に書き込みます。X バインドを有効にし、バインドされた例外として RX_FRM_ACCEPTED を Table 14: Exceptions summary から選択します。

注

表「例外の概要」(セクション 16)および例外の GPIO 構成(セクション 12.6)の番号のオフセットに注意してください。

これは、ピンを出力として設定した場合に、例外 RF_IDLE を GPIOCTRLn.CTRLn レジスタ・ビットの GPIO ピンに送信する例です。この場合、例外 RF_IDLE には表 9: GPIO の構成に従って 0x01 の番号が割り当てられます。

RF_IDLE を命令にバインドする場合、EXCBINDX/Y1 で使用する番号は、表 14: 例外の概要に従って 0x00 になります。

15 メモリ・マップ

CC2520 の構成レジスタは、アドレス 0x000 ~ 0x07F にあります。0x080 ~ 0x0FF は、現在予約されている未使用の領域です。CC2520 には、アドレス 0x100 ~ 0x3FF にある 768 バイトの物理 RAM が含まれています。

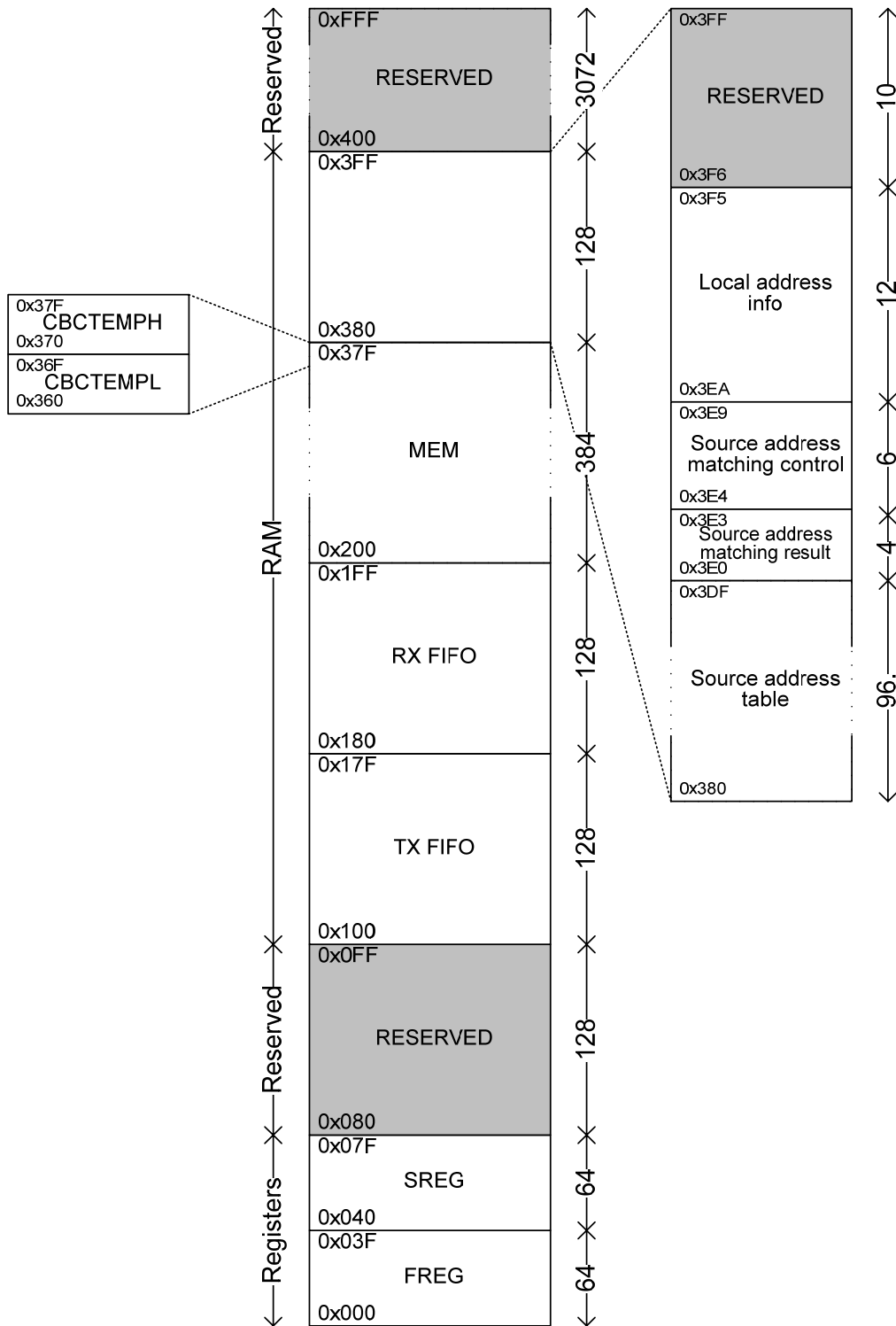


図 11: CC2520 メモリ・マップ

15.1 FREG

FREG は、REGRD および REGWR 命令を使用してアクセスできる 128 個の高速アクセス 8 ビット・レジスタです。FREG メモリ領域で開始される REGRD および REGWR 命令は、SREG まで継続でき、0x07F で折り返します。FREG は、REGRD および REGWR に関連して SPI を介して 1 つ多くのバイトを必要とする MEMRD および MEMWR 命令でもアクセスできます。

0x000 ~ 0x01F の範囲にある FREG のレジスタは、BCLR および BSET 命令を使用してビット単位で書き込むことができます。

FREG にあるレジスタについては、セクション 28 で説明しています。128 個のアドレスの一部は使用されていないことに注意してください。

15.2 SREG

SREG は、MEMRD および MEMWR 命令でアクセスできる 128 個の 8 ビット・レジスタです。

SREG にあるレジスタについては、セクション 32 で説明しています。128 個のアドレスの一部は使用されていないことに注意してください。

15.3 TX FIFO

TX FIFO メモリ領域はアドレス 0x100 ~ 0x17F にあり、128 バイトの大きさです。このメモリ領域は TX FIFO 用のものですが、まったく保護されていないため、MEMWR および MEMRD 命令でアクセスすることもできます。通常、TX FIFO の内容を操作するには、特定の命令のみを使用する必要があります。TX FIFO には一度に 1 フレームのみ含めることができます。TX FIFO の詳細については、セクション 22.3 を参照してください。

15.4 RX FIFO

RX FIFO メモリ領域はアドレス 0x180 ~ 0x1FF にあり、128 バイトの大きさです。このメモリ領域は RX FIFO 用のものですが、まったく保護されていないため、MEMWR および MEMRD 命令でアクセスすることもできます。通常、RX FIFO の内容を操作するには、特定の命令のみを使用する必要があります。RX FIFO には一度に複数のフレームを含めることができます。

15.5 MEM

アドレス 0x200 ~ 0x37F の MEM メモリ領域は、384 バイト長あります。優先順位の高低に対応した CBCTEMPH と CBCTEMPL の 16 バイトの 2 つの一時領域は、CBCMAC、UCBCMAC、CCM、および UCCM 命令で使用されます。残りの MEM 領域は、一般的な目的で使用されるメモリです。

15.6 フレーム・フィルタリングおよび送信元マッチングのメモリ・マップ

フレーム・フィルタリングおよび送信元アドレス・マッチング機能は、CC2520 の 128 バイトのメモリ・ブロックを使用して、ローカル・アドレス情報および送信元マッチングの構成と結果を格納します。このメモリ空間については、表 15 で説明しています。バイト/ワード全体ではなく、一部にのみ入力されている値は、バイト/ワードの最下位部にあります。

表 15: フレーム・フィルタリングおよび送信元マッチングのメモリ・マップ

アドレス	レジスタ/変数	エンディアン	概要
予約済み			

アドレス	レジスタ/変数	エンディアン	概要
0x3F6-3FF	Temporary storage		変数の一時的なストレージに使用されるメモリ空間。
ローカル・アドレス情報			
0x3F4-0x3F5	SHORT_ADDR		送信先アドレス・フィルタリングで使用される短いアドレス。
0x3F2-0x3F3	PAN_ID		送信先アドレス・フィルタリングで使用される PAN ID。
0x3EA-0x3F1	EXT_ADDR		送信先アドレス・フィルタリングで使用される IEEE 拡張アドレス。
送信元アドレス・マッチング制御			
0x3E9	SRCSHORTPENDEN2		24 個の短いアドレスの自動保留を有効または無効にする 8 MSB の 24 ビット・マスク。
0x3E8	SRCSHORTPENDEN1		
0x3E7	SRCSHORTPENDEN0		24 個の短いアドレスの自動保留を有効または無効にする 8 LSB の 24 ビット・マスク。
0x3E6	SRCEXTPENDEN2		12 個の拡張アドレスの自動保留を有効または無効にする 8 MSB の 24 ビット・マスク。エントリ n は SRCEXTPENDEN[2n] にマッピングされます。SRCEXTPENDEN[2n+1] のすべてのビットは無視されます。
0x3E5	SRCEXTPENDEN1		
0x3E4	SRCEXTPENDEN0		12 個の拡張アドレスの自動保留を有効または無効にする 8 LSB の 24 ビット・マスク。エントリ n は SRCEXTPENDEN[2n] にマッピングされます。SRCEXTPENDEN[2n+1] のすべてのビットは無視されます。
送信元アドレス・マッチング結果			
0x3E3	SRCRESINDEX		<p>SRCRESMASK の最下位ビット '1' のビット・インデックス、または一致する送信元がない場合は 0x3F。</p> <p>一致した場合、ビット 5 は短いアドレスでのマッチングでは '0' になり、拡張アドレスでは '1' になります。</p> <p>アクノリッジの自動保留ビット条件を満たした場合 (SRCMATCH.AUTOPEND の説明を参照)、ビット 6 は '1' になります。ビットは、アクノリッジが実際に送信されたかどうかを示すものではなく、PENDING_OR レジスタ・ビットおよび SACK/SACKPEND/SNACK ストローブにも関係ありません。</p>
0x3E2	SRCRESMASK2		24 ビット・マスクは、送信元アドレスが送信元アドレス・テーブルの個別エントリに一致することを示します。
0x3E1	SRCRESMASK1		
0x3E0	SRCRESMASK0		
短いアドレス・マッチング: エントリ panid_n + short_n で一致した場合、ビット n が SRCRESMASK に設定されます。			

アドレス	レジスタ/変数	エンディアン	概要		
			拡張アドレス・マッチング: エントリ ext_n で一致した場合、ビット 2n および 2n+1 が SRCRESMASK に設定されます。		
送信元アドレス・テーブル					
0x3DE-0x3DF	short_23	ext_11	LE	LE	2 つの短いアドレスの個別エントリ(16 ビットの PAN ID と 16 ビットの短いアドレスの組み合わせ)または 1 つの拡張アドレス・エントリ。
0x3DC-0x3DD	panid_23		LE		
0x3DA-0x3DB	short_22		LE		
0x3D8-0x3D9	panid_22		LE		

0x38E-0x38F	short_03	ext_01	LE	LE	2 つの短いアドレスの個別エントリ(16 ビットの PAN ID と 16 ビットの短いアドレスの組み合わせ)または 1 つの拡張アドレス・エントリ。
0x38C-0x38D	panid_03		LE		
0x38A-0x38B	short_02		LE		
0x388-0x389	panid_02		LE		
0x386-0x387	short_01	ext_00	LE	LE	2 つの短いアドレスの個別エントリ(16 ビットの PAN ID と 16 ビットの短いアドレスの組み合わせ)または 1 つの拡張アドレス・エントリ。
0x384-0x385	panid_01		LE		
0x382-0x383	short_00		LE		
0x380-0x381	panid_00		LE		

16 周波数とチャンネル・プログラミング

キャリア周波数は、FREQCTRL.FREQ[6:0] にある 7 ビットの周波数ワードのプログラミングによって設定されます。CC2520 では、2394MHz ~ 2507MHz の範囲内のキャリア周波数をサポートしています。キャリア周波数 F_c (単位 MHz) は $F_c = (2394 + \text{FREQCTRL.FREQ}[6:0])\text{MHz}$ で表され、1MHz ステップでプログラミングできます。

IEEE 802.15.4-2006 では、2.4GHz バンド内の 16 チャンネルを指定しています。各チャンネルは 5MHz ごとに 11 ~ 26 の番号が付けられています。チャンネル k の RF 周波数は [2] で表されます。

$$F_c = 2405 + 5(k - 11) \quad [MHz] \quad k \in [11, 26]$$

チャンネル k での動作の場合、FREQCTRL.FREQ レジスタを $\text{FREQCTRL.FREQ} = 11 + 5(k - 11)$ と設定する必要があります。

17 IEEE 802.15.4-2006 の変調形式

このセクションでは、IEEE 802.15.4-2006 で定義された 2.4GHz 直接拡散方式 (DSSS) の RF 変調形式の概要について説明します。詳細については、一般資料 [2] を参照してください。

変調および拡散機能のブロック・レベルの図は、図 12 にあります。各バイトは、それぞれ 4 ビットからなる 2 つのシンボルに分割されます。最下位シンボルは最初に送信されます。マルチバイト・フィールドの場合、最上位バイトが最初に送信されるセキュリティ関連フィールドの場合を除き、最下位バイトが最初に送信されます。

各シンボルは、それぞれ 32 個のチップで構成された 16 個の擬似ランダム・シーケンスのうちの 1 つにマッピングされます。シンボルからチップへのマッピングは、Table 16 に示されています。次に、チップ・シーケンスは 2 メガチップ/秒で送信され、最下位チップ (C₀) が各シンボルの最初に送信されます。送信されたビット・ストリームおよびチップ・シーケンスは、GPIO ピンで監視できます。GPIO での監視の構成方法については、表 9 を参照してください。

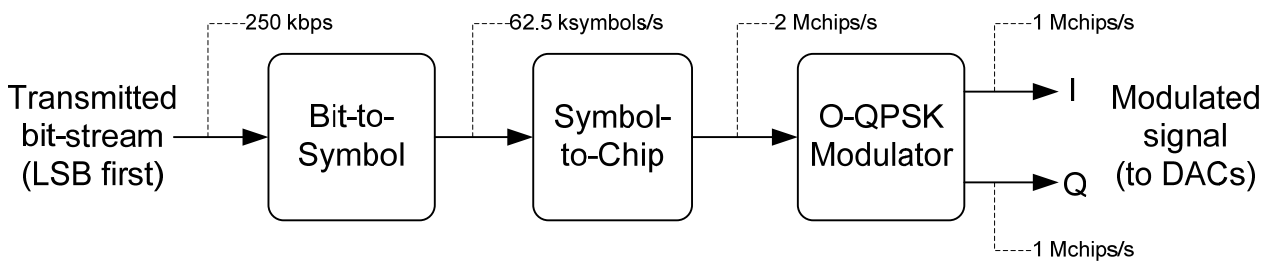


図 12: 変調

表 16: IEEE 802.15.4-2006 シンボルからチップへのマッピング

シンボル	チップ・シーケンス (C ₀ , C ₁ , C ₂ , ..., C ₃₁)
0	11011001110000110101001000101110
1	11101101100111000011010100100010
2	00101110110110011100001101010010
3	00100010111011011001110000110101
4	01010010001011101101100111000011
5	00110101001000101110110110011100
6	11000011010100100010111011011001
7	10011100001101010010001011101101
8	10001100100101100000011101111011
9	10111000110010010110000001110111
10	01111011100011001001011000000111
11	01110111101110001100100101100000
12	00000111011110111000110010010110
13	01100000011101111011100011001001
14	10010110000001110111101110001100
15	11001001011000000111011110111000

変調形式は、ハーフサインのチップ・シェーピングのオフセット四位相偏移変調 (O-QPSK) です。これは MSK 変調です。各チップはハーフサインの形状で、0.5 のチップ間隔オフセットで I チャンネルと Q チャンネルで交互に送信されます。ゼロシンボルの波形を 図 13 に示します。

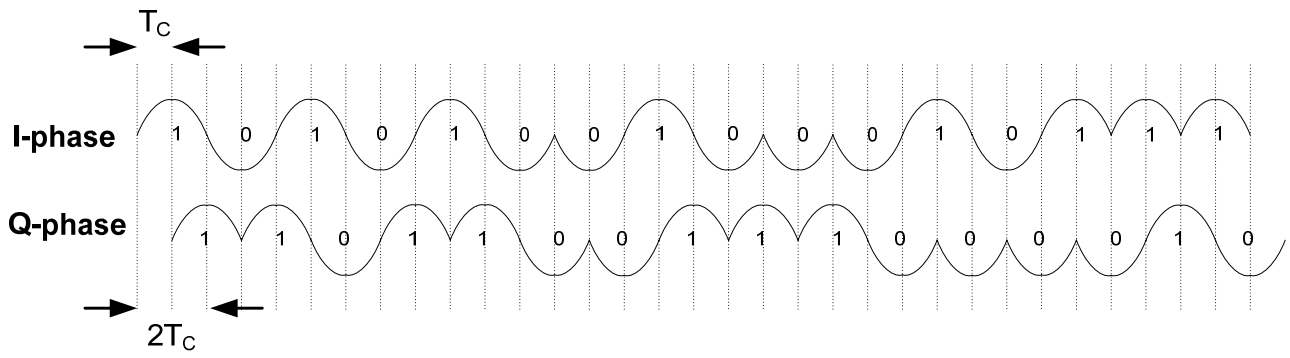


図 13: ゼロシンボルのチップ・シーケンスを送信する I / Q フェーズ、 $T_c = 0.5\mu s$

18 IEEE 802.15.4-2006 フレーム・フォーマット

このセクションでは、IEEE 802.15.4 フレーム・フォーマットに関する簡単な概要を示します [2]。CC2520 には、フレームの一部を処理するためのサポートが内蔵されています。この点について、以下のセクションで説明します。

Figure 18 に、IEEE 802.15.4 フレーム・フォーマットの概略図を示します。特定のフレーム・フォーマット(データ・フレーム、ビーコン・フレーム、および MAC コマンド・フレーム)について示す同様の図が、標準ドキュメントに含まれます [2]。

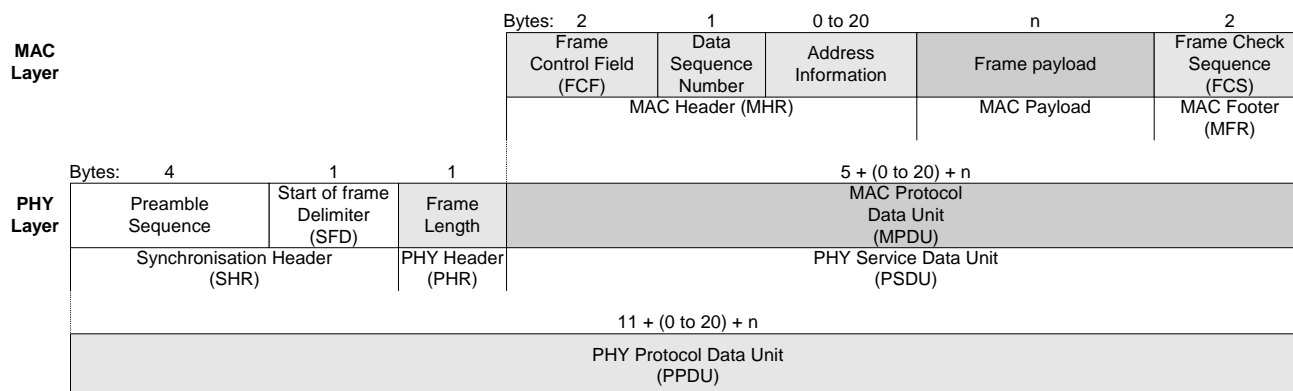


図 14: IEEE 802.15.4 フレーム・フォーマットの概略図 [1]

18.1 PHY 層

同期ヘッダ

同期ヘッダ (SHR) は、プリアンプル・シーケンスと、それに続くフレーム開始部 (SFD) によって構成されています。IEEE 802.15.4 の仕様 [2] では、プリアンプル・シーケンスは 4 バイトの 0x00 として定義されています。SFD は、1 バイトで、値 0xA7 です。

PHY ヘッダ

PHY ヘッダは、フレーム長フィールドのみで構成されています。フレーム長フィールドは、MPDU のバイト数を定義します。バイト長フィールドの値には、バイト長フィールド自体は含まれないことに注意してください。一方、FCS (フレーム・チェック・シーケンス) は CC2520 ハードウェアによって自動的に挿入されるものですが、この値に含まれません。

フレーム長フィールドの長さは 7 ビットであり、最大値は 127 です。フレーム長フィールドの最上位ビットは予約済みで、常に 0 に設定する必要があります。

PHY サービス・データ・ユニット

PHY サービス・データ・ユニットには、MAC プロトコル・データ・ユニット (MPDU) が含まれています。MPDU の生成/解釈を行うのは MAC 層の役割ですが、CC2520 には MPDU サブフィールドのいくつかを処理するためのサポートが内蔵されています。

18.2 MAC 層

フレーム長フィールドの後には、図 14 に示すように FCF、データ・シーケンス番号、およびアドレス情報が続きます。これらを MAC データ・ペイロードおよびフレーム・チェック・シーケンスと組み合わせると、MPDU が形成されます。FCF の形式を、図 15 に示します。詳細については、IEEE 802.15.4 仕様書を参照してください [2]。

ビット: 0 ~ 2	3	4	5	6	7 ~ 9	10 ~ 11	12 ~ 13	14 ~ 15
フレー ム・タイ プ	セキュ リティ 有効	フレー ム保留	アクノリッジ 応答の要求	イン トラ PA N	予約済 み	送信先ア ドレッシング グ・モード	予約済 み	送信元ア ドレッシング グ・モード

図 15: フレーム制御フィールド(FCF)のフォーマット

フレーム・チェック・シーケンス

2 バイトのフレーム・チェック・シーケンス(FCS)は、図 14 に示すように、直前の MAC ペイロード・バイトの後に続きます。FCS は MPDU 全体を対象にして計算されます、フレーム長フィールドは FCS の一部ではありません。

[2] で定義されている FCS の多項式は、次のとおりです。

$$G(x) = x^{16} + x^{12} + x^5 + 1$$

CC2520 は、FCS の自動的な計算/検証をサポートしています。詳細は、セクション 20.3 と 22.1.3 を参照してください。

19 送信モード

このセクションでは、トランスミッタの制御方法、総合的なフレーム処理、および TX FIFO の使用方法について説明します。

19.1 TX 制御

CC2520 には、フレーム制御とステータス報告をするための多くの機能が組み込まれています。CC2520 には、マイクロコントローラが出力フレームのタイミングの正確な制御を容易に実行できるようにする機能があることに注意してください。IEEE 802.15.4/ZigBee システムには、厳格なタイミング条件があるので、このことが非常に重要です。

フレーム送信は、次のアクションによって開始されます。

- STXON コマンド・ストロブ
 - SAMPLED_CCA 信号は更新されません。
- STXONCCA コマンド・ストロブ。CCA 信号が "HIGH" の場合に使用されます。
 - 処理中の送信/受信を中止し、TX キャリブレーションを強制的に実行し、その後で送信を行います。
 - SAMPLED_CCA 信号が更新されます。

クリア・チャンネル・アセスメントの詳細については、セクション 19.7 で説明します。

フレーム送信は、次のコマンド・アクションによって中止されます。

- SRXON コマンド・ストロブ
 - 処理中の送信を中止し、RX キャリブレーションを強制的に実行します。
- SRFOFF コマンド・ストロブ
 - 処理中の送信/受信を中止し、FSM を強制的に IDLE 状態にします。
- STXON コマンド・ストロブ
 - 上記を参照してください。

STXON を使用して送信を行った後にレシーバを有効にするには、FRMCTRL1.SET_RXENMASK_ON_TX ビットを設定する必要があります。この結果、STXON を実行するときに、RXENABLE のビット 14 が設定されます。STXONCCA を使用して送信をする場合、送信を行う前にレシーバがオンになり、送信後に再びオンに戻ります（その間に RXENABLE レジスタがクリアされた場合を除く）。

19.2 TX 状態のタイミング

STXON または STXONCCA コマンド・ストロブから 192 μ s 後にプリアンプルの送信が開始されます。これを [2]では「TX ターンアラウンド・タイム」と呼んでいます。受信モードに戻るときも、同じ長さの遅延が発生します。

アイドル・モードまたは受信モードに戻るときに、変調器が DAC に対して立ち下がりを実行する間、2 μ s の遅延が発生します。（フレーム長バイトで定義された）MPDU 全体を送信した後、または TX アンダーフローが発生した場合に、この立ち下がりが自動的に発生します。これは、次のことに影響を及ぼします。

- SFD 信号は、2 μ s にわたって延長されます。
- IDLE 状態への無線の FSM 遷移は 2 μ s 遅延します。

19.3 TX FIFO アクセス

TX FIFO は、128 バイトを保持できますが、1 度に 1 つのフレームのみです。TX コマンド・ストロブ実行前または実行後に、フレームをバッファリングできますが、その結果 TX アンダーフローを発生させないことが条件です（以下のエラー条件を参照してください）。

図 16 に、TX FIFO に何を書き込む必要があるのか示します(青の網掛け)。TX オーバーフローが発生しない場合は、追加のバイトは無視されます(以下のエラー条件を参照してください)。

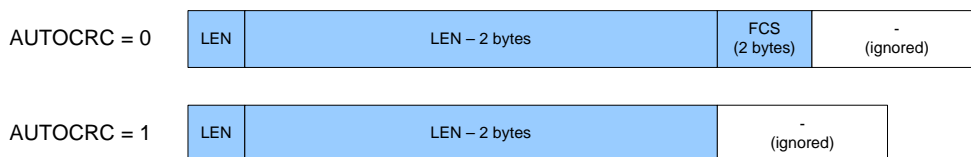


図 16. TX FIFO に対して書き込まれるフレーム・データ

TX FIFO に書き込む方法は 3 通りあります。

- TXBUF 命令を実行すると、マイクロコントローラから、CC2520 の TX FIFO に対してバイトが送信されます。
- TXBUFCP 命令を実行すると、CC2520 内の汎用 RAM から TX FIFO にバイトがコピーされます。
- フレーム・バッファリングはいつでも、TX FIFO メモリの先頭から開始されます。FRMCTRL1.IGNORE_TX_UNDERF ビットを有効にすることにより、MEMWR、MEMCP、および他のメモリ命令を実行してフレームを書き込むことも可能です。ただし、専用の TXBUF または TXBUFCP 命令を使用する方法を推奨します。

TX FIFO 内のバイト数は、TXFIFOCNT レジスタに格納されます。

SFLUSHTX コマンド・ストローブにより、TX FIFO を手動で空にすることができます。送信の間に FIFO が空にされた場合、TX アンダーフローが発生します。

19.3.1再送信

フレームの単純な再送信をサポートするために、TX FIFO の内容が送信された時点で、CC2520 は TX FIFO の内容を削除しません。フレームの送信に成功した後、FIFO の内容は変更されないままです。同じフレームをもう一度送信するには、STXON または STXONCCA コマンド・ストローブを発行することにより、単に TX を再起動します。

他のフレームを送信する場合、単に新しいフレームを TX FIFO に書き込んでください。この場合、実際の書き込みを実行する前に、TX FIFO は自動的にフラッシュされます。

19.3.2エラー条件

TX FIFO に関してエラー条件が2つあります。

- TX FIFO がフルの状態のときに残りのバイトを書き込もうとし、オーバーフローが発生しました。
- TX FIFO が空のときに CC2520 が送信の目的で次のバイトをフェッチしようとし、アンダーフローが発生しました。

TX オーバーフローは、TX_OVERFLOW 例外によって示されます。このエラーが発生した場合、書き込みは中止され、オーバーフローを発生させたデータ・バイトは失われます。SFLUSHTX ストローブにより、エラー条件をクリアする必要があります。

TX アンダーフローは、TX_UNDERFLOW 例外によって示されます。このエラー条件が発生した場合、処理中の送信は中止されます。SFLUSHTX ストローブにより、エラー条件をクリアする必要があります。

FRMCTRL1.IGNORE_TX_UNDERF ビットを設定することにより、TX_UNDERFLOW 例外を無効にすることもできます。この場合、最初のバイト(フレーム長バイト)で指定された数のバイトの送信が完了するまで、CC2520 は TX FIFO でその後に登場するバイトの送信を続けます。

19.4 TX フロー・チャート

図 17 では、フロー・チャートの形でこれまでのセクションについて要約します。

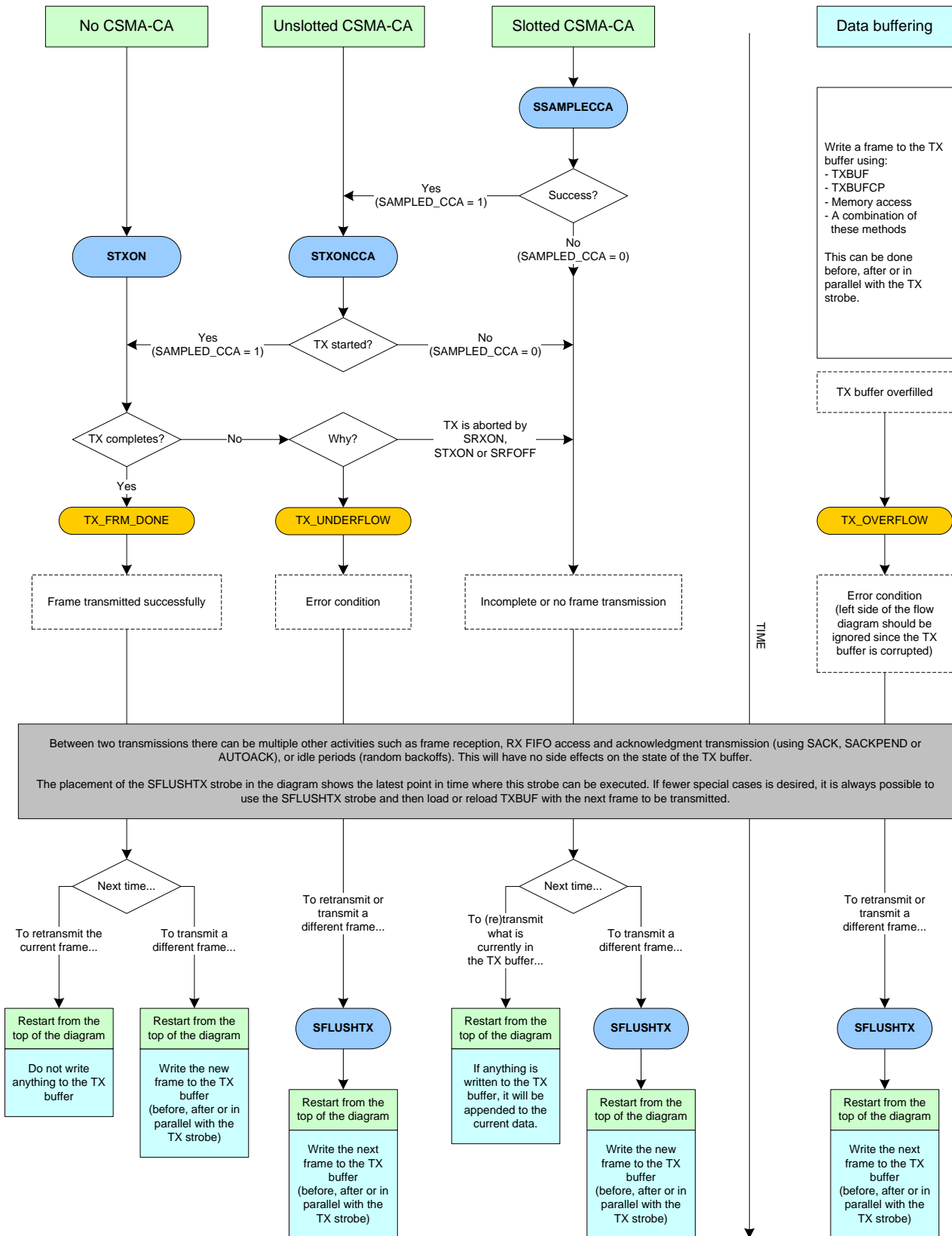


図 17: TX フロー

19.5 フレーム処理

CC2520 は、TX フレームに関して次のフレーム生成タスクを実行します。



1. プリアンブルと SFD からなる PHY 層の同期ヘッダの生成と自動的な送信。
2. フレーム長フィールドによって指定されたバイト数の送信。
3. FCS の生成と自動的な送信(無効にすることも可能)。

推奨される方法は、フレーム長フィールドの後に、MAC ヘッダ、MAC ペイロードの順に TX FIFO への書き込みを行い、残りの処理を CC2520 に任せることです。長さ 2 バイトの FCS は CC2520 によって自動的に追加されますが、これらをフレーム長フィールドに含める必要があることに注意してください。

19.5.1 同期ヘッダ

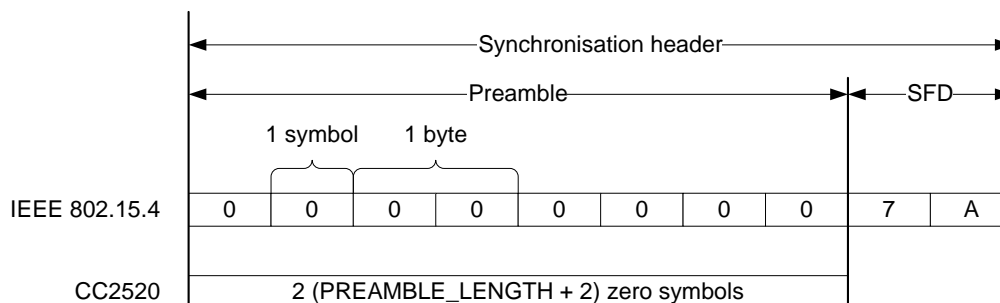


図 18: 送信される同期ヘッダ

CC2520 ではプリアンブルの長さはプログラマブル(可変)です。デフォルト値は [2] に準拠しており、この値を変更した場合、システムは IEEE 802.15.4 に準拠しなくなります。

プリアンブルのシーケンス長は、MDMCTRL0.PREAMBLE_LENGTH を使用して設定します。図 18 に、CC2520 の同期ヘッダと IEEE 802.15.4 仕様書の関連について示します。

プリアンブルの必要なバイト数を送信した後、CC2520 は長さ 1 バイトの SFD を自動的に送信します。SFD は固定されており、ソフトウェアからこの値を変更することはできません。

19.5.2 フレーム長フィールド

SFD を送信した後、CC2520 内の変調器は TX FIFO からのデータの読み取りを開始します。変調器は、フレーム長フィールド、およびそれに続く MAC ヘッダと MAC ペイロードを検出できることを予期しています。フレーム長フィールドは、送信対象となるバイト数を決定する目的で使用します。

AUTOCRC='1' の場合は最小のフレーム長が 3、AUTOCRC='0' の場合は 1 になることに注意してください。

19.5.3 フレーム・チェック・シーケンス

制御ビット FRMCTRL0.AUTOCRC が設定されている場合、FCS フィールドは CC2520 によって自動的に生成され、送信されるフレームのうち、フレーム長フィールドによって定義される位置に対して追加されます。FCS は TX FIFO に書き込まれることはありませんが、個別の 16 ビット・レジスタに格納されます。デバッグ目的などを除

き、AUTOCRC を常に有効にしておくことを推奨します。FRMCTRL0.AUTOCRC='0' の場合、変調器は TX FIFO で FCS を検出できることを予期するので、ソフトウェアは FCS を生成して MPDU の残りの内容とともに TX FIFO に書き込む必要があります。

FCS 計算機能に関する CC2520 のハードウェア実装については、Figure 22 に示します。詳細については、[2] を参照してください。

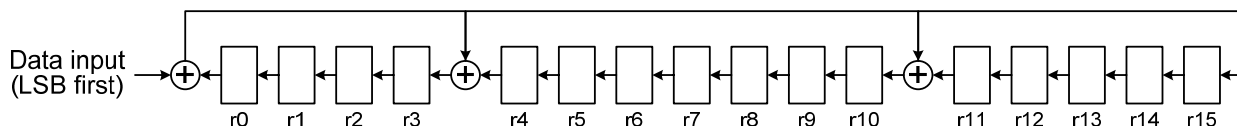


図 19: CC2520 の FCS ハードウェア実装

19.6 例外

フレームの SFD フィールドの送信が完了した時点で、SFD 例外が発生します。フレームの終了時に、フレーム全体の送信が成功した場合は TX_FRM_DONE 例外が発生します。

GPIO(構成値 0x2A)上には 2 番目の SFD 信号も利用できますが、これを SFD 例外と混同しないでください。

19.7 クリア・チャネル・アセスメント

クリア・チャネル・アセスメント(CCA)ステータス信号は、チャンネルが送信に利用できるかどうかを示します。CCA 機能を使用して、IEEE 802.15.4 仕様書 [2] で規定されている CSMA-CA 機能を実装します。レーザバが少なくとも 8 つのシンボル期間にわたって有効な状態が続いていた場合、CCA 信号が有効になります。このことを検証するために、RSSI_VALID ステータス信号を使用できます。

CCA は、RSSI 値とプログラマブル・スレッショールドに基づいています。厳密な動作は、CCACTRLO と CCACTRL1 の各レジスタ内で構成できます。

CCA 信号には 2 つの派生版があり、1 つは新しい RSSI サンプルが得られるたびに更新され、もう 1 つは SSAMPLECCA と STXONCCA の各コマンド・ストローブによってのみ更新されます。これらは両方とも FSMSTAT1 レジスタ内で利用できます。

RSSI_VALID 信号がセットされてから 4 クロック・サイクル後に CCA 信号が更新されることに注意してください。

19.8 出力電力のプログラミング

CC2520 の RF 出力電力は、TXPOWER レジスタのうち 7 ビットの値によって制御されます。Table 17 に、中心周波数を 2.440GHz に設定する場合の推奨設定に関する代表的な出力電力と消費電流を示します。これらの推奨設定は、利用可能なレジスタ設定の内の小規模なサブセットに過ぎません。表 17 に示す設定以外の設定を使用すると、消費電流が増大して、全般的な電力効率が低下します。オプションの温度補正された TX の詳細については、セクション 5.11 を参照してください。

表 17: CC2520 リファレンス・デザインを測定した出力と消費電流(+3.0V, +25°C, $f_c = 2.440\text{GHz}$ 時)

TXPOWER 設定 (hex)	出力 Typ (dBm)	消費電流 Typ (mA)
F7	5	33.6
F2	3	31.3
AB	2	28.7
13	1	27.9
32	0	25.8
81	-2	24.9
88	-4	23.1
2C	-7	19.9
03	-18	16.2

18.9 ヒントとテクニック

- GPIO ピンから STXON および STXONCCA ストローブをトリガします。この結果、マイクロコントローラは出力フレームのタイミングを非常に正確に制御できるようになります。
- マイクロコントローラ内のタイマを使用して SFD 例外のタイミングをキャプチャします。この結果、フレームの送信がいつ完了したのかマイクロコントローラが正確に把握できるようになります。
- 送信を開始する前は、TX FIFO にフレーム全体を格納する必要はないことに注意してください。送信中に、TX FIFO に対してバイトを追加することもできます。
- MDMTEST1.MODULATION_MODE='1' に設定する方法により、IEEE 802.15.4 に準拠しないフレームを送信するよう CC2520 に指示することも可能です。

20 受信モード

このセクションでは、レシーバの制御方法、総合的な RX フレーム処理、および RX FIFO の使用方法について説明します。

20.1 RX 制御

CC2520 のレシーバをオンまたはオフにするには、SRXON と SRFOFF の各コマンド・ストローブ、および RXENABLE レジスタを使用します。これらのコマンド・ストローブは「ハード」のオン/オフ・メカニズムを提供し、RXENABLE による操作は「ソフト」のオン/オフ・メカニズムを提供します。

レシーバをオン(有効)にするには、以下のアクションを実行します。

- SRXON ストローブ
 - RXENABLE[15] をセットします。
 - 遷移を強制的に RX キャリブレーションに移行することにより、処理中の送信/受信を中止します。
- STXON のストローブ。FRMCTRL1.SET_RXENMASK_ON_TX が有効になっている場合。
 - RXENABLE[14] をセットします。
 - 送信が完了した後、レシーバは有効になります。
- RXENABLE != 0x0000 に設定します。
 - 処理中の送信/受信が中止されることはありません。

レシーバをオフ(無効)にするには、以下のアクションを実行します。

- SRFOFF ストローブ
 - RXENABLE[15:0] をクリアします。
 - 遷移を強制的に IDLE モードに移行することにより、処理中の送信/受信を中止します。
- RXENABLE = 0x0000 に設定します。
 - 処理中の送信/受信が中止されることはありません。処理中の送信/受信が完了した段階で、CC2520 は IDLE 状態に戻ります。

RXENABLE レジスタを操作する方法は複数あります。

- REGWR 命令と MEMWR 命令
- BSET 命令と BCLR 命令
- RXENABLEAND 命令と RXENABLEOR 命令
- SRXMASKBITSET ストローブと SRXMASKBITCLR ストローブ (RXENABLE[13] に影響する)
- SRXON ストローブ、SRFOFF ストローブ、STXON ストローブ (FRMCTRL1.SET_RXMASK_ON_TX 設定を含む)

20.2 RX 状態のタイミング

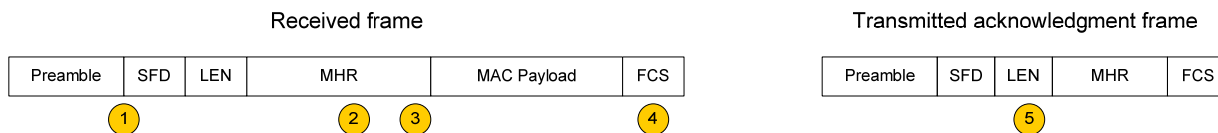
前述の方法のいずれかで RX を有効にしてから 192 μ s 後に、レシーバの準備が完了します。これを [2] では「RX ターンアラウンド・タイム」と呼んでいます。

フレーム受信後に受信モードに戻るときに、デフォルトでは 192 μ s の間隔が生じますが、この間に SFD の検出が無効になります。FSMCTRL.RX2RX_TIME_OFF をクリアすることにより、この間隔を無効にすることもできます。

20.3 フレーム処理

CC2520 は、IEEE 802.15.4-2003 および IEEE 802.15.4-2006 に記載されている RX 要件のうち重要な部分をハードウェアに統合しています。この結果、マイクロコントローラの割り込みレートが低下し、フレーム受信を処理するソフトウェアを簡素化し、最小限のレイテンシで結果を提供することができます。

単一フレームを受信している間に、CC2520 は以下のフレーム処理ステップを実行します。



1. 受信した PHY 同期ヘッダ (プリアンブルと SFD) の検出と削除、およびフレーム長フィールドで指定されたバイト数の受信。
2. [1] と [2] のセクション 7.5.6.2「third filtering level (3 番目のフィルタリング・レベル)」で規定された、フレーム・フィルタリング。
3. 最大 24 個の短いアドレスまたは最大 12 個の拡張 IEEE アドレスを保持しているテーブルと、送信元アドレスのマッチング。送信元アドレス・テーブルは、オンチップ RAM に格納されます。
4. 自動的な FCS チェックを実行し、その結果と他のステータスの値 (RSSI、LQI、および送信元のマッチング結果) を、受信済みフレームに追加。
5. 送信元アドレス・マッチング結果と FCS のチェック結果に基づき、正しいタイミングで送信に対するアクノリッジ応答を行い、フレーム保留ビットを正しく設定。

20.3.1 同期ヘッダとフレーム長フィールド

フレーム開始部 (SFD) と、その後続く、受信がいつ完了するのか決定するフレーム長バイトを検出した時点で、フレーム受信が開始されます。GPIO4 のデフォルト出力は SFD 信号ですが、この信号をマイクロコントローラのタイマ入力に接続して、受信したフレームの開始時期をキャプチャできます。

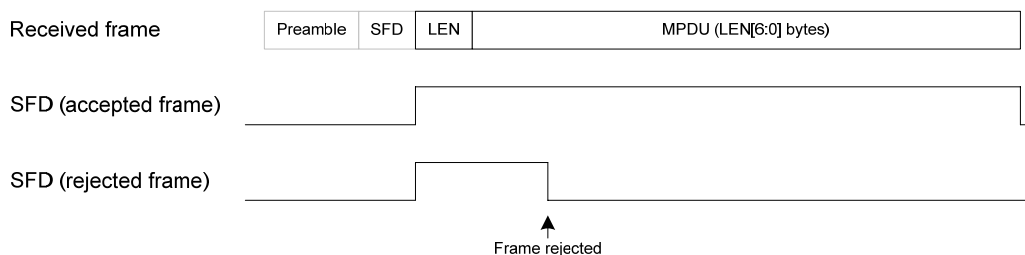


図 20: SFD 信号のタイミング

プリアンブルと SFD は RX FIFO に書き込まれません。

CC2520 は相関器を使用して SFD を検出します。MDMCTRL1.CORR_THR 内の相関スレッシュホールド値によって、受信した SFD が「理想的な」SFD にどれほど類似している必要があるのかが決定されます。このスレッシュホールド値を調整するときは注意が必要です。

- 過度に高く設定した場合、CC2520 は実際の SFD を多数見逃し、実質的にレシーバの感度を低下させることになります。
- 過度に低く設定した場合、CC2520 は多数の SFD ではないものを SFD として誤検出することになります。この場合、レシーバの感度は低下しませんが、実質的な結果は同じです。誤検出したフレームが、実際のフレームの SFD に重複する可能性があるからです。また、FCS としては正しい値を保持している、フレームではないものを、誤ってフレームとして受信するリスクも高くなります。

SFD の検出に加えて、SFD を検出する前に、規定された数の有効なプリアンブル・シンボル (相関スレッシュホールドも上回っている) を要求することも必要です。利用可能なオプションと推奨設定については、MDMCTRL0 レジスタと MDMCTRL1 レジスタに関する説明を参照してください。

CC2520 rev. A の場合、デフォルトの相関スレッシュホールドは低すぎるので、(RX を試行する前に) リセット後に更新する必要があります。

20.3.2 フレーム・フィルタリング

フレーム・フィルタリング機能により、[1] と [2] のセクション 7.5.6.2「third filtering level」(3 番目のフィルタリング・レベル)で規定されているとおり、意図されていないフレームは拒否されます。さらに、この機能は以下のものに対するフィルタリングも実施します。

- 8 つの異なるフレーム・タイプ (FRMFILT1 レジスタを参照)
- フレーム制御フィールド (FCF) 内の予約済みビット

この機能は、以下のものによって制御されます。

- FRMFILTO レジスタと FRMFILT1 レジスタ
- RAM 内にある LOCAL_PAN_ID 値、LOCAL_SHORT_ADDR 値、および LOCAL_EXT_ADDR 値

フィルタリング・アルゴリズム

FRMFILTO.FRْم_FILTER_EN ビットは、フレーム・フィルタリングを適用するかどうかを制御します。この機能が無効な場合、CC2520 は受信したすべてのフレームを受け入れます。この機能が有効な場合 (デフォルト設定)、CC2520 は以下のすべての要件を満たすフレームのみを受け入れます。

- フレーム長バイトは、「最小フレーム長」以上の値であることが必要です。「最小フレーム長」は、FCF の 送信元アドレス・モード・サブフィールド、送信先アドレス・モード・サブフィールド、PAN ID 圧縮サブフィールドから導かれます。
- FCF の予約済みのビット [9:7] と、FRMFILTO.FCF_RESERVED_BITMASK を AND 演算した結果は 0b000 であることが必要です。
- FCF のフレーム・バージョン・サブフィールドは、FRMFILTO.MAX_FRAME_VERSION を上回ることはできません。
- 送信元と送信先の各アドレス・モードは、予約済みの値 (1) にはできません。
- 送信先アドレス:
 - 送信先 PAN ID がフレームに含まれている場合、その値は LOCAL_PANID に一致しているか、ブロードキャスト PAN 識別子 (0xFFFF) と同一であることが必要です。
 - 短い送信先アドレスがフレームに含まれている場合、その値は LOCAL_SHORT_ADDR とブロードキャスト・アドレスのどちらかに一致していることが必要です。
 - 拡張送信先アドレスがフレームに含まれている場合、その値は LOCAL_EXT_ADDR に一致していることが必要です。
- フレーム・タイプ
 - ビーコン・フレーム (0) は、以下の場合にのみ受け入れられます。
 - FRMFILT1.ACCEPT_FT0_BEACON = 1
 - フレーム長バイト ≥ 9
 - 送信先アドレス・モードが 0 (送信先アドレスなし)
 - 送信元アドレス・モードが 2 または 3 (つまり、送信元アドレスがフレームに含まれる)
 - 送信元 PAN ID が LOCAL_PANID に一致しているか、LOCAL_PANID が 0xFFFF に等しい
 - データ・フレーム (1) は、以下の場合にのみ受け入れられます。
 - FRMFILT1.ACCEPT_FT1_DATA = 1
 - フレーム長バイト ≥ 9
 - 送信先アドレスと送信元アドレスの一方または両方がフレームに含まれています。送信先アドレスがフレームに含まれていない場合、FRMFILTO.PAN_COORDINATOR ビットが設定されていて、送信元 PAN ID が LOCAL_PANID に等しいことが必要です。
 - アクノリッジ応答フレーム (2) は、以下の場合にのみ受け入れられます。
 - FRMFILT1.ACCEPT_FT2_ACK = 1

- フレーム長バイト = 5
- MAC コマンド・フレーム(3)は、以下の場合にのみ受け入れられます。
 - FRMFILT1.ACCEPT_FT3_MAC_CMD = 1
 - フレーム長バイト >= 9
 - 送信先アドレスと送信元アドレスの一方または両方がフレームに含まれています。送信先アドレスがフレームに含まれていない場合、FRMFILT0.PAN_COORDINATOR ビットが設定されていて、送信元 PAN ID が、受け入れようとするフレームの LOCAL_PANID に等しいことが必要です。
- 予約済みのフレーム・タイプ(4、5、6、および 7)は、以下の場合にのみ受け入れられます。
 - FRMFILT1.ACCEPT_FT4TO7_RESERVED = 1(デフォルトは 0)
 - フレーム長バイト >= 9

フィルタリングが開始される前に、以下の操作が実行されますが、RX FIFO に格納されているフレーム・データに対して影響を及ぼすことはありません。

- フレーム長バイトのビット 7 はマスクによって除外されます(無視)。
- FRMFILT1.MODIFY_FT_FILTER が 0 である可能性が低い場合、FCF のフレーム・タイプ・サブフィールドのうち MSB は、反転されるか強制的な指定によって 0 または 1 になります。

フレームが拒否される場合、そのフレーム内で SFD を誤検出することを避けるために、(フレーム長フィールドの定義に従って)拒否されるフレームの受信が完了した後でのみ、CC2520 は新しいフレームの検索を開始します。拒否されたフレームが RX オーバーフローを発生させる可能性があることに注意してください。これに該当するのは、フレームが拒否される前に RX オーバーフローが発生した場合であることに注意してください。

例外

フレーム・フィルタリングが有効になっていて、フィルタリング・アルゴリズムが受信フレームを受け入れる場合、RX_FRM_ACCEPTED 例外が生成されます。フレーム・フィルタリングが無効になっている場合や、フィルタリング結果が明らかになる前に RX_OVERFLOW または RX_FRM_ABORTED が生成された場合は、この例外は生成されません。

Figure 24 に、3 つの異なるシナリオを示します(オーバーフローと中止によるエラー条件は含めていません)。

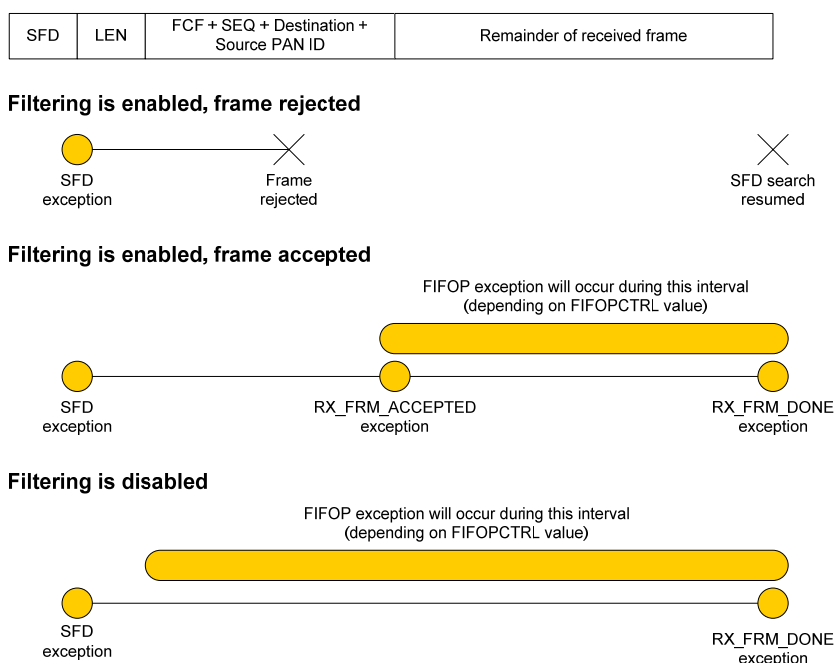


図 21: フィルタリング・シナリオ(受信時に生成される例外)

FSMSTAT1.SFD レジスタ・ビットは、フレーム開始部(SFD)の受信が完了したときに "HIGH" になり、MPDU の最後のバイトを受信するか、受信フレームがアドレス認識に失敗して拒否が完了するまでは、"HIGH" にとどまります。

SFD 例外は、GPIO ピンのみに対してルーティングすること、またはチャンネル A か B で例外のグループの一部にすることができます。SFD 例外では、マイクロコントローラのタイマ・キャプチャ・ピンに接続することを推奨します。送信および受信されるデータ・フレームの厳密なタイミング情報を抽出するためです。SFD 例外は、EXCFLAG1 レジスタにも格納されます。レジスタ・ビット(おそらく GPIO ピン)は、フレーム開始部(SFD)の受信が完了したときに "HIGH" になり、ソフトウェアによってクリアされるまで "HIGH" にとどまります。

ヒントとテクニック

以下のレジスタ設定は、正しく構成する必要があります。

- デバイスが PAN コーディネータである場合、FRMFILTO.PAN_COORDINATOR を設定する必要があります。それ以外の場合、クリアする必要があります。
- FRMFILTO.MAX_FRAME_VERSION は、IEEE 802.15.4 規格のうち、サポートされているバージョンに対応している必要があります。
- ローカル・アドレス情報を RAM にロードする必要があります。

エネルギー検出スキャンの実行中にフレームを受信することを完全に防止するには、FRMCTRL0.RX_MODE = 0b11 に設定し、RX を(再度)開始します。この結果、シンボル検索は無効になり、したがって SFD 検出は防止されます。

通常の RX モードを再開するには、FRMCTRL0.RX_MODE = 0b00 に設定し、RX を(再度)開始します。

通信量の多い IEEE 802.15.4 環境で動作する場合、CC2520 は意図していない大量のアクノリッジ応答フレームを受信することになります。これらのフレームの受信を効果的にブロックするには、FRMFILT1.ACCEPT_FT2_ACK ビットを使用して、アクノリッジ応答フレームをいつ受信するかを制御します。

- アクノリッジ応答の要求を含め、送信に成功した後、FRMFILT1.ACCEPT_FT2_ACK を設定し、アクノリッジ応答フレームを受信した後、またはタイムアウトに達した後でこのビットを再び操作してクリアします。
- それ以外の場合は、このビットをクリアしたままにします。

FRMFILTO/1 レジスタの値を変更する間、レシーバを無効にする必要はありません。ローカル・アドレス情報は RAM に格納されています。ただし、SFD バイトと送信元 PAN ID を受信する間(つまり、SFD と RX_FRM_ACCEPTED 例外の間)に変更を実施した場合は、該当のフレームに関して変更された値は「無視」とみなす必要があります(CC2520 は、古い値と新しい値のどちらかを使用します)。

MDMTEST1.MODULATION_MODE='1' に設定する方法により、すべての IEEE 802.15.4 受信フレームを無視するよう CC2520 に指示することもできることに注意してください。

20.3.3 送信元アドレス・マッチング

CC2520 は、受信フレームにある送信元アドレスを、オンチップ・メモリ内に格納されているテーブルとマッチングする機能もサポートしています。このテーブルは 96 バイトの長さであり、したがって、最大で以下のものを保持できます。

- 24 個の短いアドレス(それぞれが 2 + 2 バイト)
- 12 個の IEEE 拡張アドレス(それぞれが 8 バイト)。

送信元アドレス・マッチングは、フレーム・フィルタリングが有効であり、なおかつ受信フレームがすでに受け入れられた場合にのみ実行されます。この機能は、以下のものによって制御されます。

- SRCMATCH、SRCSHORTEN0、SRCSHORTEN1、SRCSHORTEN2、SRCEXTEN0、SRCEXTEN1、SRCEXTEN2 の各レジスタ
- RAM 内にある送信元アドレス・テーブル。

アプリケーション

フレーム保留ビットを正しく設定したアクノリッジ応答の自動送信: 間接的なフレーム送信を使用する場合、デバイスはコーディネータ内に格納されているフレームをポーリングするようデータ要求を送信します。デバイスに対応するフレームをコーディネータが実際に格納しているかどうかを示すために、コーディネータは返されるアクノリッジ応答フレームでフレーム保留ビットを設定またはクリアする必要があります。ただし、大半の 8 ビットおよび 16 ビット MCU では、このことを決定する十分な時間がありません。したがって、デバイスに対応する保留中のフレームが存在しているかどうかにかかわらず、コーディネータが最終的に保留ビットを設定することになります (IEEE 802.15.4 [2] の条件)。これは消費電力の観点では浪費を意味します。ポーリング・デバイスは、自らに対応するフレームが存在していない場合であっても、かなりの期間にわたって自らのレーザを有効なままにする必要が生じるからです。間接的なフレーム・キューにある送信先アドレスを、送信元アドレス・テーブルに対してロードし、AUTOPEND 機能を有効にすることにより、CC2520 は出力するアクノリッジ応答フレームで保留ビットを自動的に設定します。この方法により、操作に関する時間は必要ではなくなります。間接的なフレーム・キューでフレームを追加または削除するときや、送信元アドレス・テーブルをそれに応じて更新するときに、マイクロコントローラがこの作業を実施するからです。

セキュリティ関連データの参照: セキュリティで保護されたフレームの処理に要する時間を短縮するために、送信元アドレス・テーブルを設定し、マイクロコントローラ上にあるセキュリティ・キーのテーブルに対してエントリをマッチングさせることができます。テーブル・エントリに対する 2 番目のレベルのマスキングにより、このアプリケーションを、アクノリッジ応答フレームで保留ビットを自動的に設定する方法と組み合わせることができます。

他のアプリケーション: 前述の 2 つのアプリケーションは、送信元アドレス・マッチングの主要な目標です。ただし、基本的な IEEE 802.15.4 フレーム・フォーマットのみを使用する独自プロトコルの場合、役に立つ他のいくつかのアプリケーションが存在します。たとえば、そのようなアルゴリズムを例外バインド・メカニズムと組み合わせることにより、指定したノード・セットのみに対してアクノリッジ応答を行うようファイアウォール機能を作成することも可能です。

送信元アドレス・テーブル

送信元アドレス・テーブルは、図 11 に示すように、RAM のアドレス 0x380 から始まります。この領域は短いアドレスと拡張アドレスの間で共有されるので、SRCSHORTEN0/1/2 レジスタと SRCEXTEN0/1/2 レジスタを使用して、どのエントリを有効にするのかを制御します。テーブル内にあるすべての値は、リトル・エンディアンです (受信フレーム内と同様)。

- **短いアドレス・エントリ**は、16 ビットの PAN ID で始まり、その後 16 ビットの短いアドレスが続きます。これらのエントリは、アドレス $0x380 + (4 \times n)$ に格納されますが、ここで n は 0 ~ 23 の範囲にある数値です。
- **拡張アドレス・エントリ**は、64 ビットの IEEE 拡張アドレスのみで構成されています。これらのエントリは、アドレス $0x380 + (8 \times n)$ に格納されますが、ここで n は 0 ~ 11 の範囲にある数値です。

アドレス・イネーブル・レジスタ

ソフトウェアはテーブル・エントリを割り当て、アクティブな短いアドレスと拡張アドレスが重ならないようにする責任を負います。短いアドレスと拡張アドレスのそれぞれに対して、個別のイネーブル・ビットがあります。

- **短いアドレス・エントリ**は、SRCSHORTEN0、SRCSHORTEN1、SRCSHORTEN2 の各レジスタで有効になります。レジスタのビット n は、短いアドレスのエントリ n に対応します。
- **拡張アドレス・エントリ**は、SRCEXTEN0、SRCEXTEN1、SRCEXTEN2 の各レジスタで有効になります。この場合、レジスタのビット $2n$ は、拡張アドレスのエントリ n に対応します。(短いアドレスと拡張アドレスそれぞれのイネーブル・ビットからなる) 複合型のビット・ベクトルを作成して未使用のエントリを見つけるときに、このマッピングは便利に使用できます。さらに、読み取りを行うときに、レジスタのビット $2n+1$ は必ずレジスタのビット $2n$ と同じ値になります。拡張アドレスは、2 つの短いアドレス・エントリと同じメモリを占有するからです。

Memory address	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F	
0x3D8-0x3DF	ext_11								
	panid_22		short_22		panid_23		panid_23		
0x3D0-0x3D7	ext_10								
	panid_20		short_20		panid_21		short_21		SRCSHORTEN2 = 0x1C
0x3C8-0x3CF	ext_9								
	panid_18		short_18		panid_19		short_19		SRCEXTEN2 = 0xC3
0x3C0-0x3C7	ext_8								
	panid_16		short_16		panid_17		short_17		
0x3B8-0x3BF	ext_7								
	panid_14		short_14		panid_15		short_15		
0x3B0-0x3B7	ext_6								
	panid_12		short_12		panid_13		short_13		SRCSHORTEN1 = 0x10
0x3A8-0x3AF	ext_5								
	panid_10		short_10		panid_11		short_11		SRCEXTEN1 = 0x03
0x3A0-0x3A7	ext_4								
	panid_8		short_8		panid_9		short_9		
0x398-0x39F	ext_3								
	panid_6		short_6		panid_7		short_7		
0x390-0x397	ext_2								
	panid_4		short_4		panid_5		short_5		SRCSHORTEN0 = 0x0B
0x388-0x38F	ext_1								
	panid_2		short_2		panid_3		short_3		SRCEXTEN0 = 0xF0
0x380-0x387	ext_0								
	panid_0		short_0		panid_1		short_1		

図 22 - イネーブル・テーブル・エントリの例

比較アルゴリズム

SRCMATCH.SRC_MATCH_EN ビットは、送信元アドレス比較機能を有効にするかどうかを制御します。有効になっていて(これはデフォルト設定です)、フレームがフィルタリング・アルゴリズムを通過した場合、CC2520 は送信元アドレスの表現タイプに応じて、図 22 で概要を示したアルゴリズムのいずれかを適用します。

結果は 2 つの異なる形式で報告されます。

- SRCRESMASK という 24 ビットのベクトルでは、一致するものが存在する有効化された短いエントリごとに 1 個の '1' を保持し、一致するものが存在する有効化された拡張エントリごとに 2 個の '1' を保持します(読み取りアクセスを行う場合、アドレス有効化レジスタに関するビット・マッピングは共通です)。
- SRCRESINDEX という 7 ビットの値の場合：
 - 受信フレームで送信元アドレスが提示されていない場合や、受信した送信元アドレスに対して一致するものが存在しない場合：
 - ビット 6:0: 0x3F
 - 受信した送信元アドレスに対して一致するものが存在する場合：
 - ビット 4:0: 比較に使用される最初のエントリ(つまり、インデックス番号が最小のエントリ)に対応するインデックスであり、短いアドレスの場合は 0 ~ 23、拡張アドレスの場合は 0 ~ 11 です。
 - ビット 5: 短いアドレスに関して一致するものが存在する場合は '0'、拡張アドレスに関して一致するものが存在する場合は '1' です。
 - ビット 6: AUTOPEND 機能の結果です。

<p>短い送信元アドレス(モード 2) 受信した送信元 PAN ID を srcPanid と呼びます。 受信した短いアドレスを srcShort と呼びます。</p> <pre> SRCRESMASK = 0x000000; SRCRESINDEX = 0x3F; for (n = 0; n < 24; n++) { bitVector = 0x000001 << n; if (SRCSHORTEN & bitVector) {</pre>	<p>拡張送信元アドレス(モード 3) 受信した拡張アドレスを srcExt と呼びます。</p> <pre> SRCRESMASK = 0x000000; SRCRESINDEX = 0x3F; for (n = 0; n < 12; n++) { bitVector = 0x000003 << (2*n); if (SRCEXTEN & bitVector) { if (ext[n] == srcExt) { SRCRESMASK = bitVector; if (SRCRESINDEX == 0x3F) {</pre>
---	---

<pre> if ((panid[n] == srcPanid) && (short[n] == srcShort)) { SRCRESMASK = bitVector; if (SRCRESINDEX == 0x3F) { SRCRESINDEX = n; } } } } </pre>	<pre> SRCRESINDEX = n 0x20; } } } } } } </pre>
---	--

図 23 - 短いアドレスと拡張アドレスに対応する比較アルゴリズム

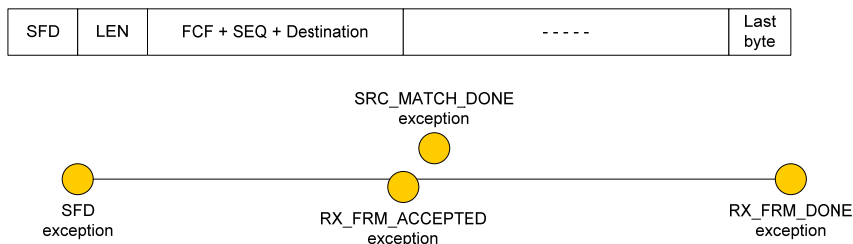
結果が利用可能になった直後に、SRCRESMASK と SRCRESINDEX は CC2520 のメモリに書き込まれます。FRMCTRL0.AUTOCRC ビットと FRMCTRL0.APPEND_DATA_MODE ビットがセットされている場合、SRCRESINDEX も受信フレームに対して追加されます。その後、この値は、16 ビットのステータス・ワードのうち 7 ビットの LQI の値を置き換えます。

例外

送信元アドレス・マッチングが有効になっていて、比較アルゴリズムが完了した場合、その結果にかかわらず SRC_MATCH_DONE の例外が生成されます。一致するものが見つかった場合、SRC_MATCH_FOUND 例外も生成され、その直後に SRC_MATCH_DONE が続きます。

図 24 に、これらの例外のタイミングを示します。

When there is no source address:



When there is a source address:

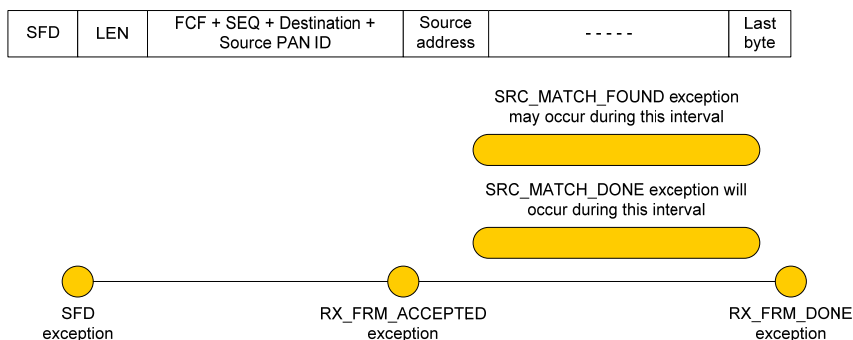


図 24 - 送信元アドレス・マッチングによって生成される例外

ヒントとテクニック

- フレーム受信の実行中に、送信元アドレス・テーブルを安全に変更できます。レシーバがアクティブである間にあるアドレスを使用して他のアドレスを置き換える場合、変更を実行するときに、対応するイネーブル・ビットを無効にする必要があります。この結果、CC2520 が古い値と新しい値を組み合わせ使用することを防止できません。CC2520 は、送信元マッチング・プロセス全体を実行している間、エントリーが有効であることのみを想定するからです。

以下の対策を実行して、次に受信するフレームが、送信元アドレス・マッチングの結果を上書きすることを防止できます。

- RAM に書き込まれた値の代わりに、追加された SRCRESINDEX の結果を使用します(これは、推奨されるアプローチです)。
- 次の受信フレームに関する RX_FRM_ACCEPTED が発生する前に、RAM から結果を読み取ります。最も短いフレーム・タイプの場合、シーケンス番号の直後に RX_FRM_ACCEPTED が発生するので、利用可能な合計時間(絶対的なワーストケースに短い安全性マージンを加えた長さ)は次のようになります。

$$16\mu s(\text{必須のプリアンブル}) + 32\mu s(\text{SFD}) + 128\mu s(4 \text{ バイト}) = 176\mu s$$

- 利用可能な時間を長くするには、FSMCTRL.RX2RX_TIME_OFF ビットをクリアします。この結果、さらに 192μs が追加され、合計 368μs になります。これは、RX オーバーフローのリスクを低減することにもなりません。

20.3.4 フレーム・チェック・シーケンス

受信モードでは、FRMCTRL0.AUTOCRC が有効になっている場合、FCS がハードウェアによって検証されます。ユーザは通常、FCS シーケンス自体ではなく、FCS の正しさのみに興味があります。したがって、受信の際に FCS シーケンス自体は RX FIFO に対して書き込まれません。代わりに、FRMCTRL0.AUTOCRC が設定されている場合、2 バイトの FCS はより役に立つ他の値で置き換えられます。FCS シーケンスに関連して、どの値を置き換えに使用するかは、FRMCTRL0 レジスタで構成可能です。

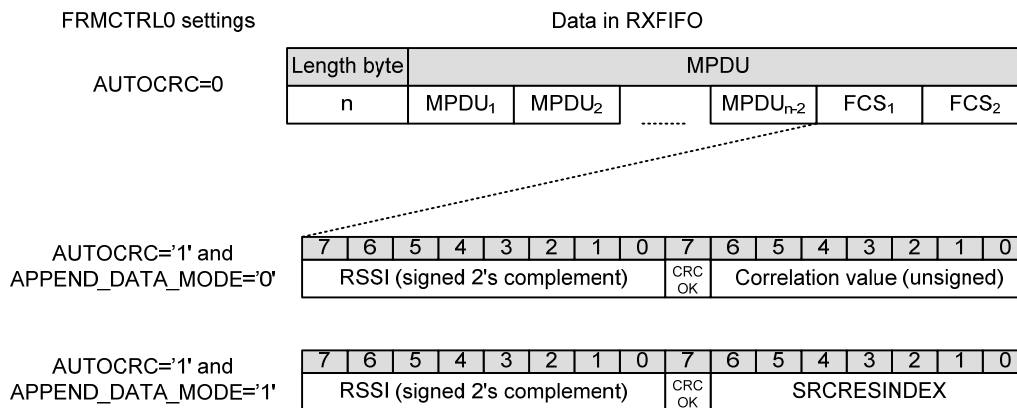


図 25: さまざまな設定に対する RX FIFO 内のデータ

フィールドの説明:

- RSSI の値は、SFD に続く最初の 8 個のシンボルにわたって測定されます。
- CRC_OK ビットは、FCS が正しいか(1)、正しくないか(0)を示します。正しくない場合、ソフトウェアはそのフレームを破棄する責任を負います。
- 相関値は、SFD に続く最初の 8 個のシンボルにわたる相関値の平均です。
- SRCRESINDEX は、送信元アドレス比較が完了した後に RAM に書き込まれたのと同じ値です。

IEEE 802.15.4 で使用される LQI 値の計算方法をセクション 20.5 で説明します。

20.3.5 アクノリッジ応答の送信

CC2520 には、フレームの受信が成功した(つまり受信フレームの FCS が正しいことが必要)後、アクノリッジ応答を送信するためのハードウェア・サポートが含まれています。図 26 に、アクノリッジ応答フレーム・フォーマットを示します。

Bytes:	4	1	1	2	1	2
	Preamble Sequence	Start of Frame Delimiter (SFD)	Frame Length	Frame Control Field (FCF)	Data Sequence Number	Frame Check Sequence (FCS)
	Synchronisation Header (SHR)		PHY Header (PHR)	MAC Header (MHR)		MAC Footer (MFR)

図 26. アクノリッジ応答フレーム・フォーマット

生成されるアクノリッジ応答フレームには、3 つの可変フィールドがあります。

- 保留ビットは、コマンド・ストローブと AUTOPEND 機能を使用して制御できます。
- データ・シーケンス番号 (DSN) は、直前に取得したフレームから自動的に取得されます。
- FCS は、暗黙に決定されます。

ACK フレーム内の保留ビットを設定するために 3 つの異なるソースがあります(つまり、SACKPEND ストローブ、PENDING_OR レジスタ・ビット、および AUTOPEND 機能)。これらのソースのいずれかがセットされた場合、保留ビットがセットされます。

送信タイミング

アクノリッジ応答フレームは、フレームを受信した直後にのみ送信できます。送信タイミングは、FSMCTRL.SLOTTED_ACK ビットによって制御されます。

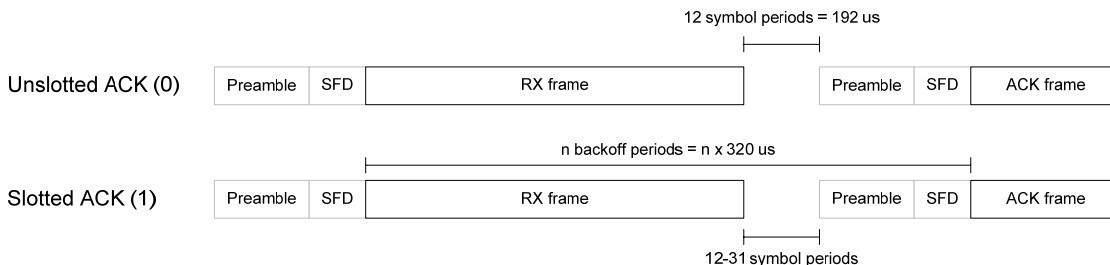


図 27: アクノリッジ応答のタイミング

802.15.4 では、ビーコン非対応 PAN では非スロット・モード、ビーコン対応 PAN ではスロット・モードが要求されています。

手動制御

SACK、SACKPEND、SNACK の各コマンド・ストローブは、フレーム受信中にのみ発行できます。それ以外のときにこれらのストローブを発行した場合、何も影響を及ぼさず、代わりに USAGE_ERROR 例外が生成されます。

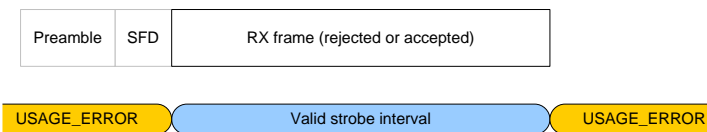


図 28: コマンド・ストローブのタイミング

コマンド・ストローブは受信中に複数回発行することもできますが、最後のストローブのみが影響を及ぼします。

- ストローブなし / SNACK / 正しくない FCS : アクノリッジ応答を送信しません
- SACK: フレーム保留ビットをクリアした状態でアクノリッジ応答を送信します
- SACKPEND: フレーム保留ビットを設定した状態でアクノリッジ応答を送信します

自動制御(AUTOACK)

FRMFILTO.FRAME_FILTER_EN と FRMCTRL0.AUTOACK の両方を有効にした状態では、CC2520 はアクノリッジ応答フレームを送信するかどうかを自動的に決定します。

- RX フレームは、フレーム・フィルタリングによって受け入れられていることが必要です (RX_FRM_ACCEPTED 例外によってその結果が示されます)
- RX フレームでアクノリッジ応答要求ビットが設定されていることが必要です
- RX フレームは、ビーコン・フレームやアクノリッジ応答フレームであってはなりません
- RX フレームの FCS は正確であることが必要です

自動的なアクノリッジ応答を、SACK、SACKPEND、SNACK の各コマンド・ストローブによって上書きすることもできます。たとえば、マイクロコントローラのメモリ・リソースが不足していて受信フレームを格納できない場合、受信中に SNACK ストローブを発行し、破棄されるフレームに対してアクノリッジ応答を行うことを防止できます。

デフォルトでは、AUTOACK 機能はアクノリッジ応答フレーム内でフレーム保留ビットを決して設定しません。コマンド・ストローブによる手動の上書き以外に、2 つのオプションもあります。

- AUTOPEND 機能を使用する自動制御
- FRMCTRL1.PENDING_OR ビットを使用する手動制御

フレーム保留フィールドの自動設定(AUTOPEND)

SRCMATCH.AUTOPEND ビットが設定されている場合、送信元アドレス比較の結果によって、フレーム保留ビットの値を決定します。フレームを受信するときに、以下の条件が成立する場合、(おそらく)返そうとするアクノリッジ応答フレーム内のフレーム保留フィールドがセットされます。

- FRMFILTO.FRAME_FILTER_EN が設定されています。
- SRCMATCH.SRC_MATCH_EN が設定されています。
- SRCMATCH.AUTOPEND が設定されています。
- 受信フレームが現在の SRCMATCH.PEND_DATAREQ_ONLY 設定と一致しています。
- 受信した送信元アドレスが、送信元マッチング・テーブル・エントリのいずれかと一致しています。このテーブルは、SRCSHORTEN と SRCSHORTPENDEN の両方、または SRCEXTEN と SRCEXTPENDEN の両方によって有効にされます。

送信元比較テーブルがいっぱいになった場合、FRMCTRL1.PENDING_OR ビットを使用して AUTOPEND 機能を上書きし、一時的にフレーム保留ビットをセットした上ですべてのフレームに対してアクノリッジ応答を送信することもできます。

20.4 RX FIFO アクセス

RX FIFO は、合計バイト数が 128 以下であることを前提として、1 つ以上の受信フレームを保持できます。RX FIFO にあるバイト数を判断するには、2 つの方法があります。

- RXFIFOCNT レジスタを読み取ります
- FIFOP と FIFO の各信号を、FIFOPCTRL.FIFOPTHR 設定と組み合わせて使用します

RX FIFO にアクセスする方法はいくつかあります。

- RXBUF 命令は CC2520 から受信したバイトをマイクロコントローラに転送します
- RXBUFCP 命令は CC2520 から受信したバイトをマイクロコントローラに転送し、CC2520 の RAM から読み取ったバイトのコピーを作成します
- RXBUFMOV 命令は RX FIFO から受信したバイトを CC2520 の RAM に転送します
- ソフトウェアは RXFIRST レジスタを使用して、RX FIFO にある先頭バイトを取得できます

SFLUSHRX コマンド・ストローブは RX FIFO をリセットし、すべての受信フレームを削除し、すべてのカウンタ、ステータス信号、および固定的なエラー条件をクリアします。

20.4.1 FIFO 信号と FIFOP 信号の使用方法

FIFO 信号と FIFOP 信号は、フレームを受信している最中に受信フレームの一部を読み出すのに役立ちます。

- RX FIFO に 1 バイト以上の内容が存在しているときに FIFO 信号 (GPIO1 に対するデフォルトの出力) は "HIGH" になりますが、RX オーバーフローが発生したときに "LOW" になります。
- FIFOP 信号 (GPIO2 に対するデフォルトの出力) は、以下の場合に "HIGH" になります。
 - RX FIFO にある有効なバイト数が、FIFOPCTRL にプログラムされた FIFOP のスレッシュホールド値を上回っている場合。フレーム・フィルタリングが有効になっていて、フレームの受信が完了するまでフレーム・ヘッダー内のバイトが有効であるとみなされていない場合。
 - FIFOP スレッシュホールドに達していない状況も含め、新しいフレームの最後のバイトを受信した場合。FIFOP スレッシュホールドに達していない状況では、次に RX FIFO に読み取りアクセスするときに FIFOP は "LOW" に戻ります。

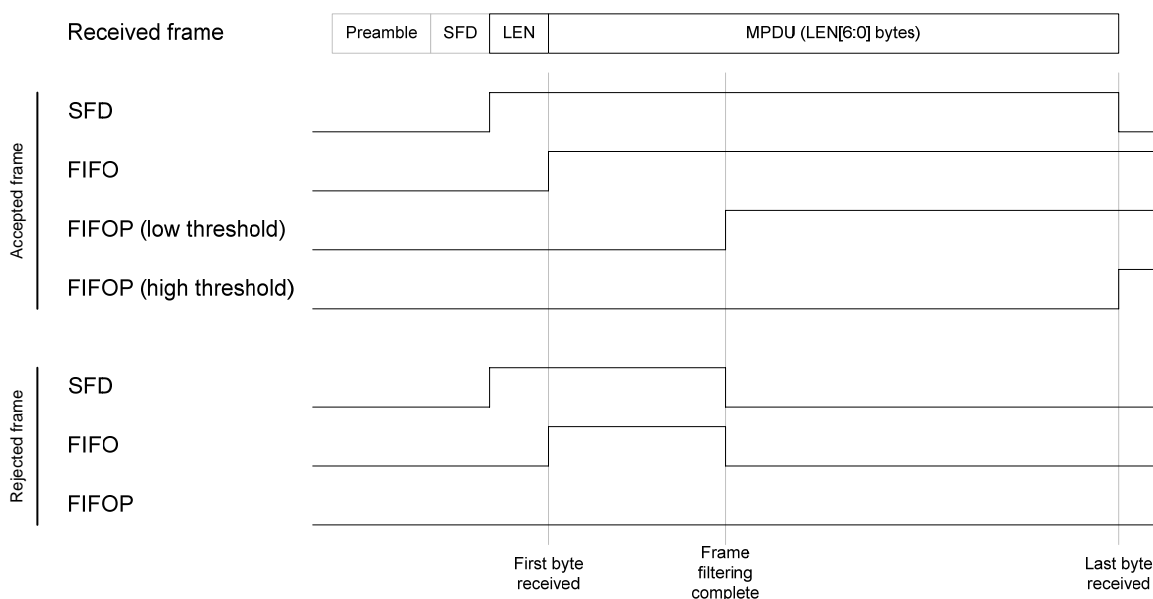


図 29: FIFO 信号と FIFOP 信号の動作。

FIFOP 信号をマイクロコントローラに対する割り込み信号として使用する場合、次の割り込みの準備をするために、割り込みサービス・ルーチンを使用して FIFOP スレッシュホールドを調整する必要があります。フレームに対する前回の割り込みに合わせて準備をするときに、スレッシュホールドを残りのバイト数に一致させる必要があります。

20.4.2 エラー条件

RX FIFO に関連するエラー条件が2つあります。

- オーバーフロー。RX FIFO がいっぱいになるときに他のバイトを受信した場合です
- アンダーフロー。ソフトウェアが空の RX FIFO からバイトを読み取ろうとした場合です

RX オーバーフローは、RX_OVERFLOW 例外と、FIFO = 0 および FIFOP = 1 という信号の値によって示されます。このエラーが発生した場合、フレーム受信が停止されます。現在 RX FIFO に格納されているフレームは、SFLUSHRX ストローブによってこの条件がクリアされるまでは読み出すことができます。フレームが拒否される前にこの条件が発生した場合、拒否されたフレームが RX オーバーフローを発生させる可能性があることに注意してください。

RX アンダーフローは、RX_UNDERFLOW 例外によって示されます。RX アンダーフローは重大なエラー条件であり、エラーフリーのソフトウェアを使用している場合は発生してはならないものです。RX_UNDERFLOW 例外は、

デバッグまたは「ウォッチドッグ」機能を目的としている場合にのみ使用する必要があります。新しいバイトの受信と同時に読み取り操作を実行している場合、RX_UNDERFLOW 例外が生成されないことに注意してください。

20.5 RSSI

CC2520 には RSSI (Received Signal Strength Indication、受信信号強度) 機能を持っています、これは 8 ビットの符号付きデジタル値を計算しますが、この値はレジスタから読み取るか、受信フレームに対して自動的に追加することができます。IEEE 802.15.4 [2] で規定されているように、RSSI 値は 8 個のシンボル期間 (128µs) にわたって、受信した電力を平均した結果です。

RSSI 値は、1dB ステップの対数スケールで、符号付き数値を 2 の補数で表現したものです。

RSSI 値レジスタを読み取る前に、ステータス・ビット RSSI_VALID をチェックする必要があります。RSSI_VALID は、レジスタ内の RSSI 値が本当に有効であるかどうか、つまり、少なくとも 8 個のシンボル期間にわたってレシーバが継続して有効になっていたかどうかを表します。

RF ピンにおける実際の信号電力 P を妥当な精度で求めるには、RSSI 値に対してオフセットを追加する必要があります。

$$P = \text{RSSI} - \text{OFFSET} [\text{dBm}]$$

オフセットは経験的な値であり、特性評価を行うときに得られるもので、CC2520 のリファレンス・デザインでは約 76dBm です。たとえば、RSSI レジスタから読み取った RSSI 値が -10 である場合、RF 入力電力は約 -86dBm になります。

RSSI レジスタが最初に有効になった後、CC2520 が RSSI レジスタを更新する方法は構成可能です。FRMCTRL0.ENERGY_SCAN='0' (デフォルト) の場合、RSSI レジスタは利用可能な最新の値を保持しています。一方、このビットが '1' に設定されている場合、ピーク検索が実行され、RSSI レジスタには、エネルギー・スキャンが有効になったとき以来、最大の値が格納されます。

20.6 リンク品質情報

LQI (リンク品質情報) は、IEEE 802.15.4 規格 [2] で定義されている、受信フレームの強度や品質に関する測定値です。LQI 値は IEEE 802.15.4 規格 [2] によって、0 ~ 255 の範囲に制限すること、および少なくとも 8 個の一意の値を示すことが要求されています。CC2520 は LQI 値を直接提供することはありませんが、マイクロコントローラが LQI 値を計算するために使用できるいくつかの測定値を報告します。

MAC ソフトウェアは RSSI 値を使用して LQI 値を計算できます。このアプローチには欠点があります。たとえば、チャンネル帯域幅の内側にナローバンドの干渉が存在する場合、本当のリンク品質は低下しているにもかかわらず RSSI が大きくなり、LQI 値も大きくなります。したがって、SFD に続く最初の 8 個のシンボルに基づき、CC2520 は各受信フレームに対して平均相関値を提供します。この符号なし 7 ビットの値は「チップ・エラー・レート」の測定値とみなすことができますが、CC2520 はチップ判定を行いません。

セクション 20.3.4 で説明したように、MDMCTRL0.AUTOCRC が設定されている場合、最初の 8 個のシンボルに対応する平均相関値が、RSSI および CRC OK/not OK とともに各受信フレームに対して追加されます。110 に近い相関値は最高品質のフレームを表し、50 に近い値は通常は CC2520 が検出可能な最低品質のフレームを表します。

ソフトウェアは [2] で規定されているように、相関値を 0 ~ 255 の範囲に変換する必要があります。たとえば、次の計算を行います。

$$LQI = (\text{CORR} - a)b$$

この式で 0 ~ 255 の範囲への制限を行うには、PER 測定値を相関値の関数とみなして a と b を経験的に見つけることができます。

RSSI と相関値の組み合わせを使用して、LQI 値を生成することもできます。

21 無線制御ステート・マシン

FSM モジュールは TX FIFO ポインタと RX FIFO ポインタを維持する責任を負い、電源投入/シャットダウンのようなアナログの「動的な」信号の制御、RF コア内のデータ・フローの制御、アクノリッジ応答フレームの自動的な生成、すべてのアナログ RF キャリブレーションの制御を行います。

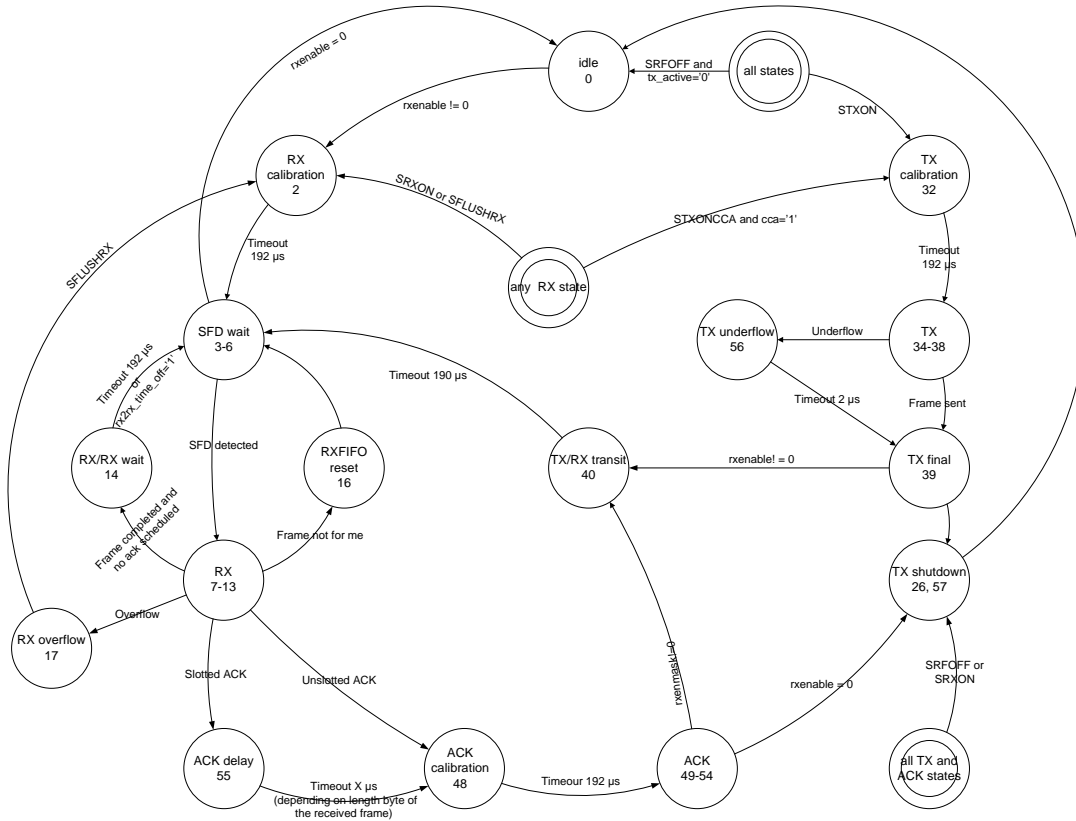


図 30: メイン FSM

表 18 に、FSM の状態から、FSMSTAT0 レジスタから読み取ることができる各数値へのマッピングを示します。FSM の状態を読み取ることは可能ですが、アプリケーション・ソフトウェア内でプログラムのフローを制御する目的でこの情報を使用することを避けてください。これらの状態は非常に迅速に変化し(32MHz のクロック・サイクルごと)、8MHz の SPI はすべてのアクティビティをキャプチャすることはできません。

表 18: FSM 状態マッピング

状態名	状態番号 10 進	番号 16 進	tx_active	rx_active
アイドル	0	0x00	0	0
RX キャリブレーション	2	0x02	0	1
SFD 待機	3 - 6	0x03 - 0x06	0	1
RX	7 - 13	0x07 - 0x0D	0	1
RX/RX 待機	14	0x0E	0	1

状態名	状態番号 10 進	番号 16 進	tx_active	rx_active
RXFIFO リセット	16	0x10	0	1
RX オーバーフロー	17	0x11	0	0
TX キャリブレーション	32	0x20	1	0
TX	34 - 38	0x22 - 0x26	1	0
TX 最終	39	0x27	1	0
TX/RX 切り替え	40	0x28	1	0
ACK キャリブレーション	48	0x30	1	0
ACK	49 - 54	0x31 - 0x36	1	0
ACK 遅延	55	0x37	1	0
TX アンダーフロー	56	0x38	1	0
TX シャットダウン	26, 57	0x1A, 0x39	1	0

22 クリスタル・オシレータ

内蔵のクリスタル・オシレータはメイン周波数リファレンスを生成します。リファレンス周波数は、32MHz であることが必要です。クリスタルの周波数は、データ・レートおよび他の内部信号処理機能に対するリファレンスとして使用されているので、他の周波数を使用することはできません。

クリスタルは、XOSC32M_Q1 ピンと XOSC32M_Q2 ピンの間に接続する必要があります。オシレータは、クリスタルのパラレル・モード動作を想定して設計されたものです。さらに、クリスタルに対する負荷コンデンサ (C121 と C131) も必須です。負荷コンデンサの値は、クリスタルに対して規定されている総負荷容量 C_L によって決まります。クリスタルの端子間に存在している総負荷容量は、クリスタルが規定の周波数で発振するための C_L と同じ値であることが必要です。CC2520 には、オシレータの周波数を最適化するために容量を追加するための機能もあります。FREQTUNE レジスタを使用すると、追加の容量の大きさを構成できます。

$$C_L = C_{tune} + C_{parasitic} + \frac{1}{\frac{1}{C_{121}} + \frac{1}{C_{131}}}$$

寄生容量はピンの入力容量と PCB の浮遊容量によって形成されます。総寄生容量は通常、2pF ~ 5pF です。

FREQTUNE レジスタのデフォルト値が「追加の容量なし」を意味していることに注意してください。これは、周波数の低下のみが可能であることを表します。外付けのコンデンサ (C121 と C131) を減らすことにより、デフォルト周波数を高くすることも可能です。この方法で、目標周波数を中心として正と負両方の調整を行って実際の周波数チューニング・レンジを移動することも可能です。

クリスタル・オシレータは、振幅制御されています。これは、オシレータを起動するために大電流が使用されることを意味します。振幅が形成された時点で、安定的な発振を維持するために必要なレベルまで電流が減少します。この結果、高速な起動が保証されると同時に、駆動電流のレベルを最小限に抑えることができます。信頼性の高い起動を保証するために、クリスタルの ESR (等価直列抵抗) は規定の範囲内にとどまっている必要があります。

クリスタル固有のパラメータ (チューニングを含む) の詳細については、セクション 6 を参照してください。

23 外部クロック出力

CC2520 は、50-50 デューティ・サイクルのクロック信号を外部回路に供給できます。これは、できるだけ部品点数を減らすことが望まれる低コスト・システムにとって利点になります。マイクロコントローラと CC2520 の両方が同じクリスタルを使用して動作できるからです。CC2520 にはクロック・ディバイダが内蔵されていて、1MHz ~ 16MHz の範囲で多くの異なる周波数の間でグリッチ・フリーの変更を実現することができます。

リセット後、CC2520 は 1MHz のクロックを GPIO0 に出力できます。クリスタル・オシレータを動作させて、外部クロックを供給するには CC2520 をアクティブ・モードにしておく必要があることに注意してください。

LPM2 からシステムをウェークアップさせる手順は、以下のとおりです。

- MCU が RC オシレータを使用して動作していて、CC2520 は LPM2 モードにあります
- CC2520 を LPM2 からアクティブ・モードに変更します。
- MCU を、CC2520 が GPIO0 に出力する 1MHz のクロックに切り替えます。
- 希望のクロック周波数に変更します。

システムをアクティブ・モードから LPM2 に切り替えるための手順は、以下のとおりです。

- MCU を RC オシレータに切り替えます。
- CC2520 を LPM2 に設定します。

24 乱数生成

CC2520 は 2 つの方法でランダム・ビットを出力できます。両方に共通するのは、ランダム・ビットを生成する必要がある場合、チップを RX にする必要があることです。遷移が完了するまで、チップを十分長い時間にわたって RX にとどめておく必要があります。これを実行するための便利な方法は、RSSI の有効な信号が "HIGH" になるまで待機することです。

- I チャンネルまたは Q チャンネル(構成可能)から取得した単一のランダム・ビットは、8MHz のレートで GPIO ピンに対して出力することができます。GPIO ピンにビットを出力する前に、I のビットと Q のビットの XOR 演算を実行する方法を選択することもできます。これらのビットは、復調器にあるデシメーション・フィルタを経由した後、I チャンネルと Q チャンネルの一方または両方にある最下位ビット(LSB)から取得したものです。
- CC2520 は、RANDOM という名前の命令をサポートしていますが、これを使用するとユーザはランダムに生成されたバイトを SPI 経由で読み取ることができます。これらのバイトは復調器内のチャンネル・フィルタから出力された I チャンネルの最下位ビット(LSB)から生成されます。

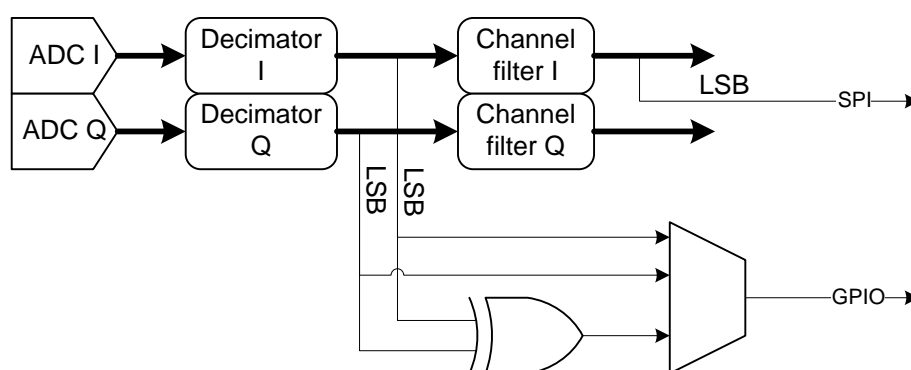


図 31: 復調器内内のランダム・ビット生成

RANDOM 命令の簡単なテストを行ったところ、大半の実用的な用途で満足のパフォーマンスが得られることが明らかになりました。RANDOM 命令を使用して約 2,000 万バイトを読み取りました。0 ~ 255 の範囲の符号なし整数として解釈したところ、平均値は 127.6518 でしたが、これは DC 成分が存在していることを示しています。

最初の 2^{14} バイトに関する FFT の結果を Figure 33 に示します。DC 成分が明確に識別できることに注意してください。2,000 万個の値に対応するヒストグラム(32 個のビン)を Figure 34 に示します。

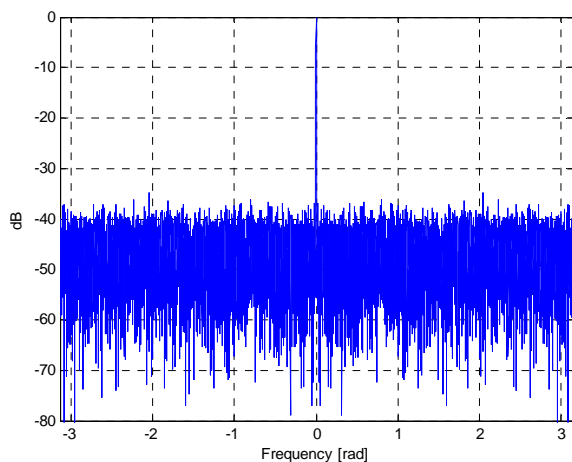


図 32: ランダム・バイトの FFT

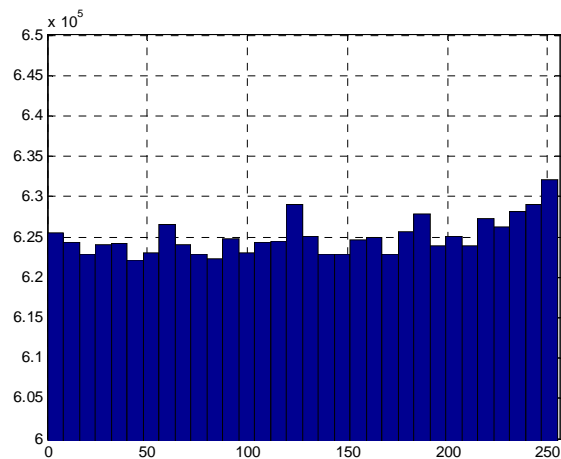


図 33: RANDOM 命令を使用して生成した 2,000 万バイトのヒストグラム

最初の 2,000 万個の個別のビットに関して、1 である確率は $P(1)=0.500602$ 、 $P(0)=1-P(1)=0.499398$ です。

乱数生成機能が「本当に無作為である」ことを完全に立証するには、より入念なテストが必要であることに注意してください。この点で役に立つ可能性のあるソフトウェア・パッケージがインターネットから入手できます [8][9]。

25 メモリ管理命令

CC2520 にはメモリの内容を管理するための命令がいくつかあります。これらの命令は、ユーザがメモリの内容を柔軟性のある方法で操作できるようにサポートされているものであり、その結果 SPI トラフィックが最小限に削減されます。これらの命令は、セキュア・フレームの操作をするときに特に役立ちます。

これらの命令に対応するパラメータは、入力データと出力データのブロックが重なり合う形で値を設定できることに注意してください。MEMCPR を除き、このことは他のすべての命令にとって何の問題ももたらしません。MEMCPR は、C = 16 の場合に限り、入力/出力が重なり合っても動作します。たとえば、9 バイトのデータ・ブロックを 1 バイト上にシフトする場合、MEMCP 命令を MEMCP(P=0,C=9,A=0x22A,E=0x22B) のように使用できます。

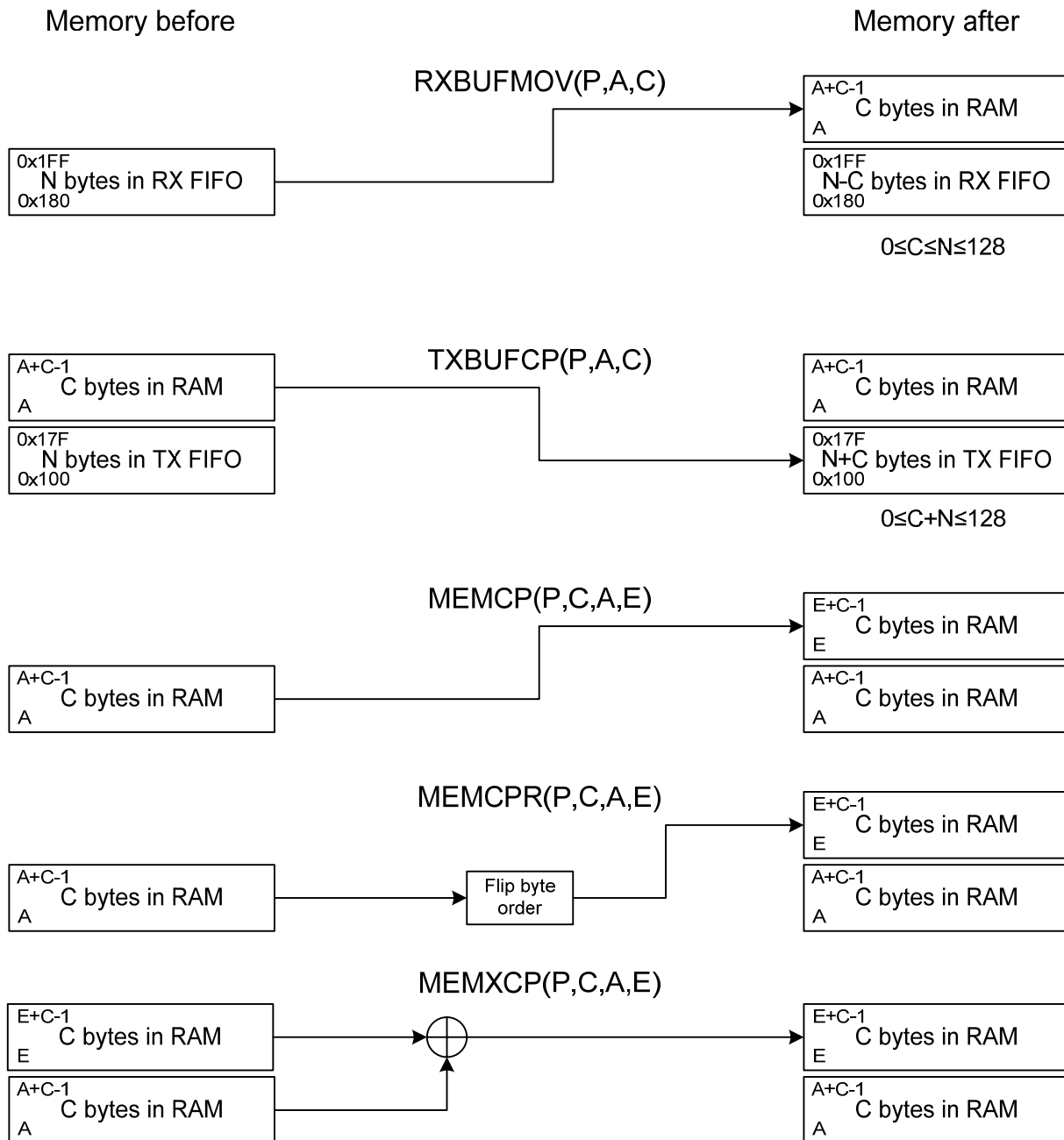


図 34: メモリ管理命令の図示

25.1 RXBUFMOV

RXBUFMOV 命令は、Figure 35 に示すように、RX FIFO からデータを読み取り、メモリ内の指定された場所にそのデータを格納します。

25.2 TXBUFCP

TXBUFCP 命令は、Figure 35 に示すように、メモリ内の指定された場所で始まるデータ・ブロックを TX FIFO にコピーします。

25.3 MEMCP

MEMCP 命令は、Figure 35 に示すように、メモリ内の指定された場所で始まるデータ・ブロックを、メモリ内の他の場所にコピーします。

25.4 MEMCPR

MEMCPR 命令は、メモリ内の指定された場所で始まるデータ・ブロックを、エンディアンを反転した上でメモリ内の他の場所にコピーします。つまり、メモリ内の場所 $A+n$ にあるバイトは、メモリ内の場所 $E+C-1-n$ にコピーされます。

25.5 MEMXCP

MEMXCP 命令は、2 つのデータ・ブロックの XOR を実行し、その結果をメモリ内のうち 2 番目のブロックに相当する位置に書き戻します。この命令は主に、セキュリティ命令に対応するサブルーチンとして使用されます。単一の短い SPI 命令で RAM のブロックをクリア (0 に設定) する目的で、この命令を使用することもできます。コピー元とコピー先という 2 つのアドレスに対して同一のアドレスを指定すると、データ自体に対する XOR が実行され、その結果、0 が RAM に書き戻されます。

26 セキュリティ命令

CC2520 は、IEEE 802.15.4 で定義されているセキュリティ処理を幅広くサポートしています。仕様書の最新バージョン [2] では、CCM* モードの処理のみが規定されています。CCM* は機密性を維持するために CTR モードの暗号化を使用し、認証の目的で CBC-MAC を使用します。CC2520 では、これらの処理を個別の命令という形で利用できます。基本的な ECB 命令が利用できるほか、CTR モードの暗号化/復号化で使用されるカウンタを操作するための命令もあります。

IEEE802.15.4 のさまざまなセキュリティ処理すべてで、AES 128 ビット暗号化のみを使用することに注意してください。復号化は決して使用されないため、CC2520 は暗号化のみをサポートしています。ECB 復号化はサポートされていません。

すべてのセキュリティ命令に関して、RAM 内でキーとカウンタをデータとは逆のバイト順序で格納する必要があることに注意してください。RAM にキー/カウンタを書き込む前にバイト順序を反転するか、MEMCPR 命令を使用してすでに RAM 内に存在しているキー/カウンタのバイト順序を反転することができます。

26.1 CC2520 内の Flags フィールドのデコード

このセクションでは、カウンタ・モードの暗号化と CBC-MAC モードの認証(CCM* も含む)との間に使用されるセキュリティ・フラグについて、および CC2520 RAM でフラグが表現される方法について定義します。

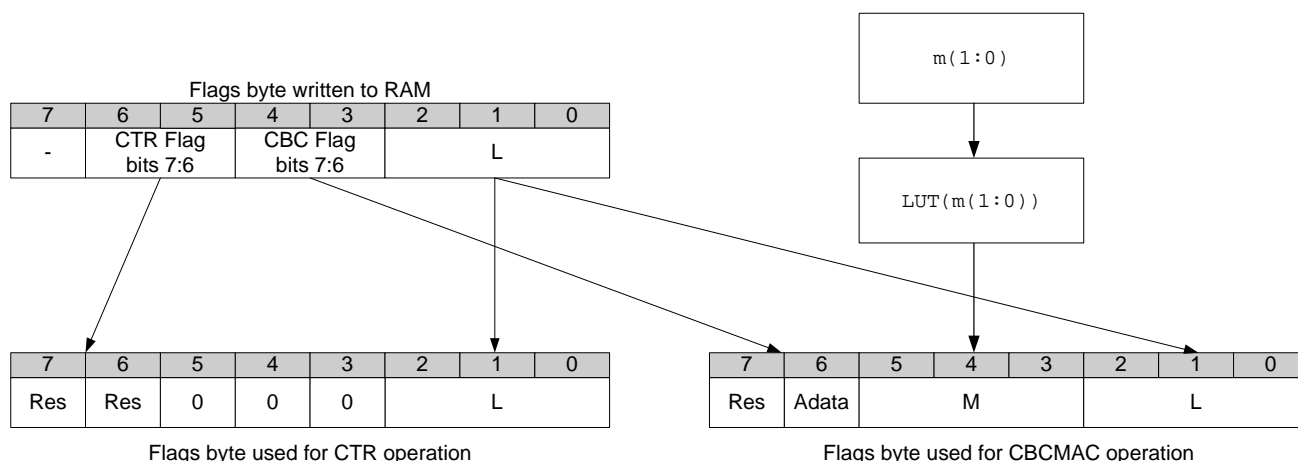


図 35: セキュリティ・フラグ

Figure 36 に、CC2520 RAM 内のカウンタの最上位バイトで、CTR と CBC-MAC のセキュリティ・フラグが表現される方法を示します。

CBC-MAC フラグの場合、参照手順を使用して、CBCMAC 命令に対する m パラメータの最下位 2 ビットを、フラグ・バイト内で使用される値 M に変換します。CCM 命令の CBC-MAC 部分に対して、同じ変換が使用されます。

表 19: 値 M に対応する参照テーブル

m(1:0)	M
00	000
01	001
10	011
11	111

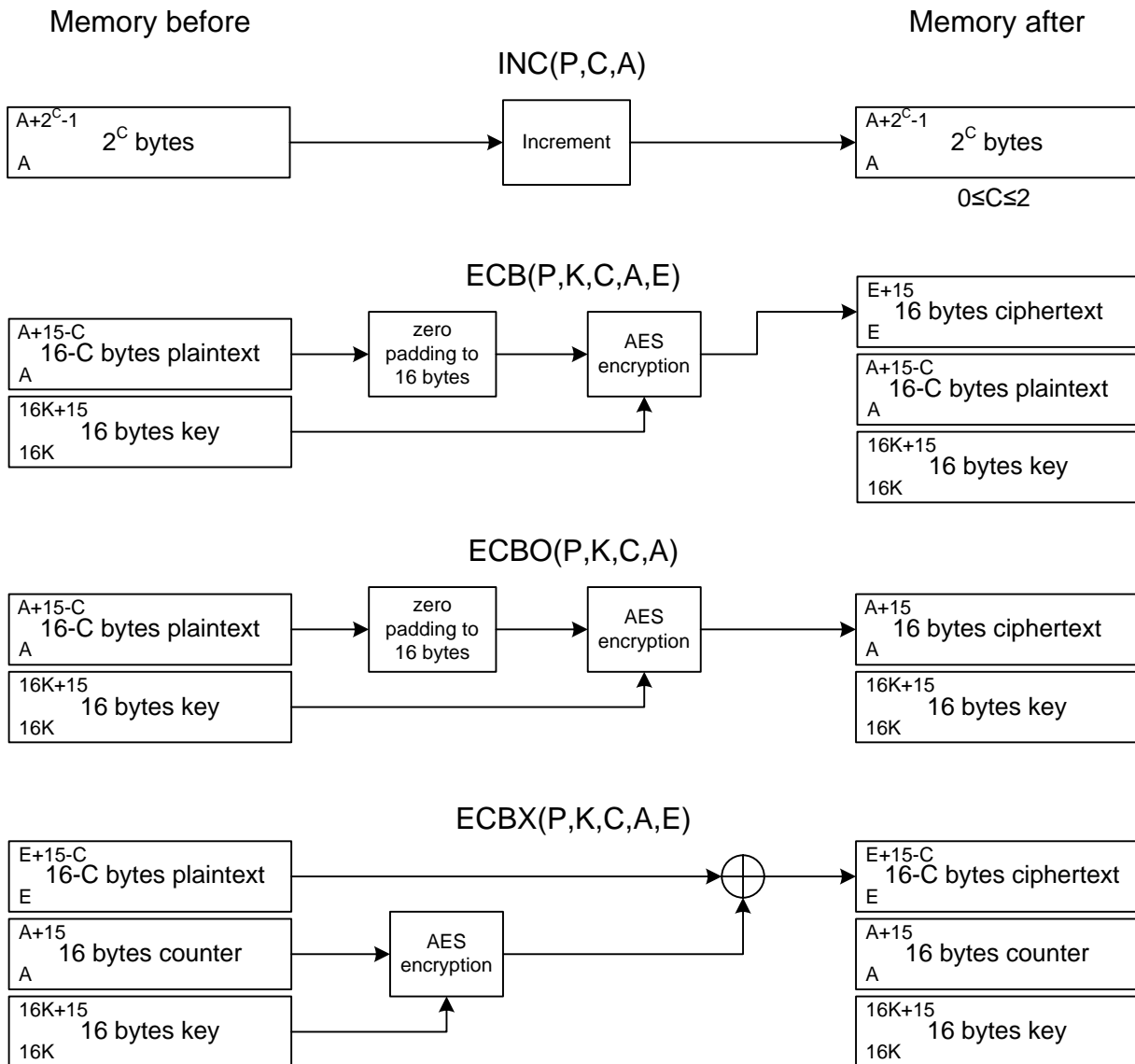


図 36: 単純な暗号化命令

26.2 INC

INC 命令は、アドレス A を LSB(最下位ビット)として、1 バイト、2 バイト、または 4 バイトのインクリメント(増加)を実行します。 $C=3$ はパラメータの値として無効であることに注意してください。

26.3 ECB

ECB 命令は、基本的なブロック暗号化を実行します。この命令は主に、CBC-MAC や CCM などより複雑な命令によって、関数として使用されることを意図しています。復号化命令が存在しないので、ECB 自体はあまり役に立ちません。暗号化テキスト出力を元に戻すことはできません。このことを欠点とみなさないでください。ECB ブロックの暗号化/復号化は、通信の安全な形態とは考えられていないからです。

この命令が、メモリのうちすでに使用されているセクションを上書きすることがないように、パラメータ E と C の値は注意深く選択する必要があります。 $E=A$ の場合、ECB 命令は ECBO とまったく同じ動作をします。

26.4 ECBO

ECBO 命令は、入力を読み取るために使用したのと同じメモリ上の場所に出力を格納することを除き、ECB と同じ動作をします。この命令は、入力が 16 バイト未満である場合でも、必ず 16 バイトを出力します。

26.5 ECBX

ECBX 命令は、暗号化の結果と、出力アドレスとして指定されたメモリの内容の間でビット単位の XOR 演算を実行することを除き、ECB と同じ動作をします。

この命令が、メモリのうちすでに使用されているセクションを上書きすることがないように、パラメータ E と C の値は注意深く選択する必要があります。E=A の場合、ECBX 命令は ECBXO とまったく同じ動作をします。

カウンタ・モードの暗号化に関する用語が Figure 37 で使用されていることに注意してください。

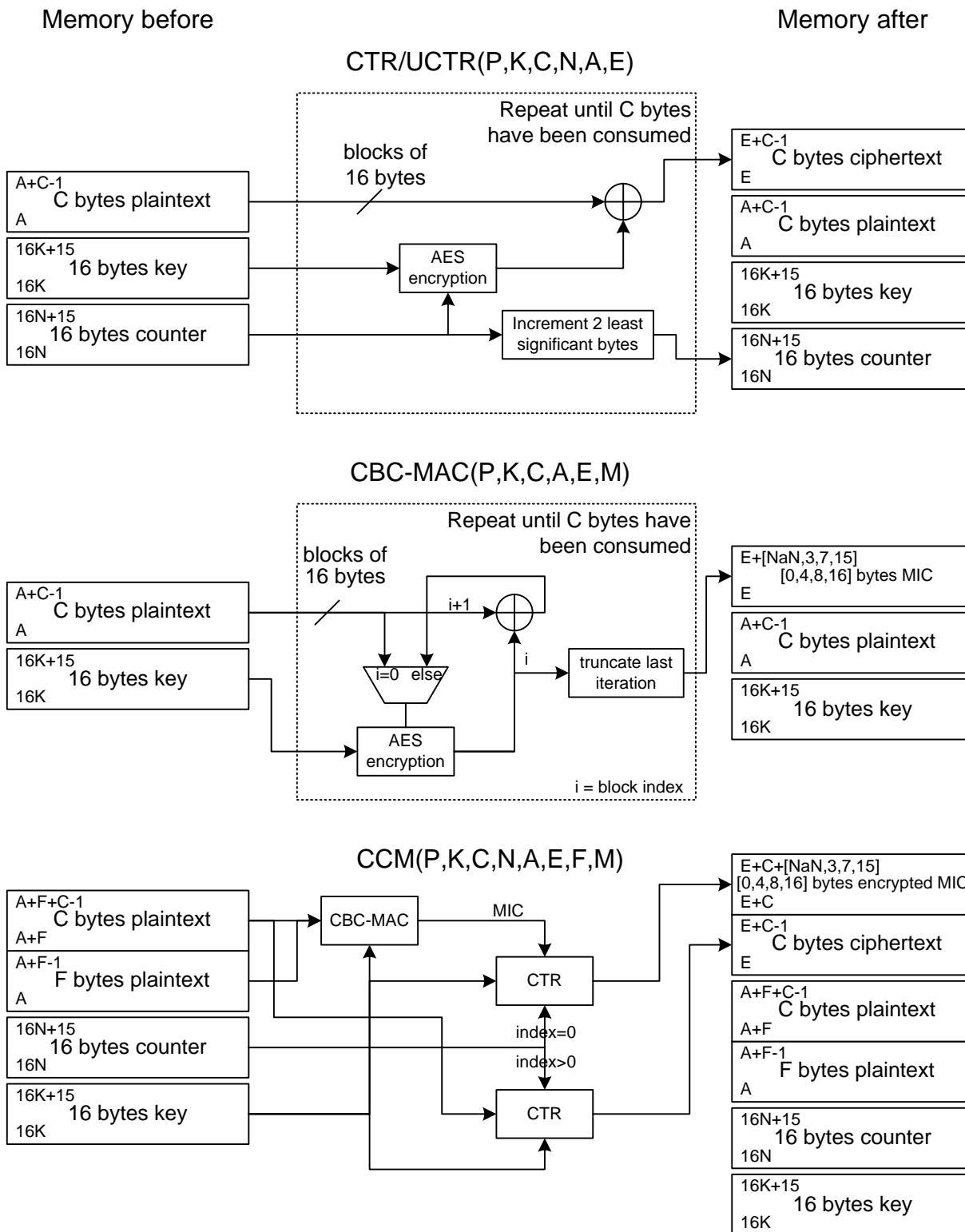


図 37: 高度なセキュリティ命令

26.6 CTR / UCTR

CTR 命令は、構成可能な長さ C であるプレーンテキスト・バイトに対して、カウンタ・モードの暗号化を実行します。この命令は、長さ C の暗号文バイトを出力します。カウンタのうち最下位 2 バイトは、プレーンテキストのうち 16 バイト・ブロックの処理が完了するたびにインクリメントされます。

プレーンテキストの最後のブロックが 16 バイトの長さでない場合、暗号化から得られたビットのうち必要な数だけ (MSB 側から順に取得したもの) が XOR 演算に使用されます。

UCTR 命令はカウンタ・モードの復号化を実行しますが、その結果は CTR 命令と必ず同一になります。カウンタ・モードの暗号化と復号化は対称型の操作だからです。

26.7 CBC-MAC

CBC-MAC 命令は、認証を実行します。

$M[1:0]=0$ の場合、認証コードが出力されないことに注意してください。 $M[1:0]$ の値がこれとは異なる場合、出力される認証バイトの数は $2^{M[1:0]+1}$ になります。 $M[2]=0$ の場合、プレーンテキスト・データに対して、C の値を 8 ビットに拡張して MSB 側に 0 を追加したものがプリフィックスとして追加されます。

26.8 CCM / UCCM

CCM 命令は CBC-MAC と CTR の両方を使用して、認証と暗号化の両方を実行します。この命令は IEEE 802.14.5-2006 [2] で規定されているように、CCM* モードの処理をサポートしています。

認証 (CBC-MAC) 部分は、 $A \sim A+F+C-1$ のアドレス範囲に対してメッセージ整合性コード (MIC) を計算します。その結果得られた MIC は、インデックス 0 のカウンタ値を使用し、CTR モード暗号化によって暗号化されます。

暗号化 (CTR) 部分は、CTR モード暗号化とインデックス 1 およびそれ以降のカウンタ値を使用して、 $A+F \sim A+F+C-1$ のアドレス範囲を暗号化しますが、その結果、C バイトの暗号文が生成されます。

暗号文と暗号化済み MIC を連結した出力は、アドレス E で始まるメモリに書き込まれます。

UCCM 命令は CTR モード復号化を使用して、 $A+F \sim A+F+C-1$ のアドレス範囲にある暗号文を復号化します。その後、CCM 命令に関連して MIC が同じ方法で生成され、入力データに含まれている MIC と比較されます。MIC の比較結果は、DPUSTAT レジスタに格納されます。

26.8.1 CCM 命令と UCCM 命令に対する入力

CCM 命令と UCCM 命令に対する入力は、図 38 と図 39 で詳細に説明されていますが、IEEE 802.15.4 仕様書で使用されている用語との関連に関する注釈もあります。

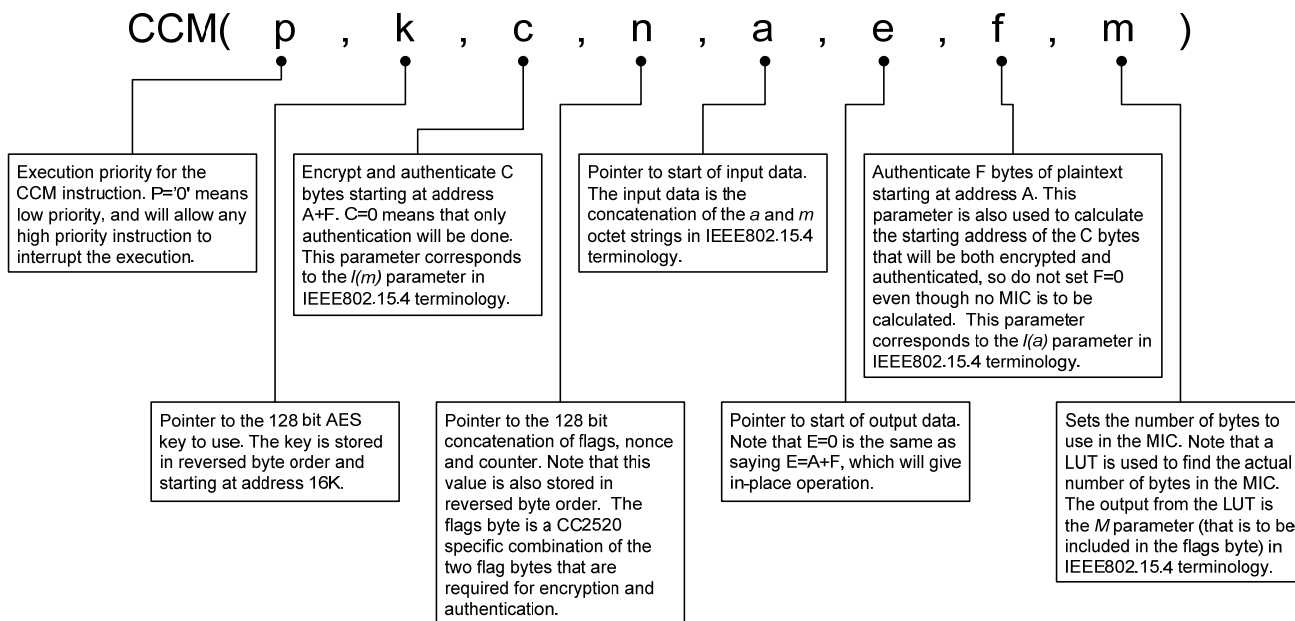


図 38: CCM 命令に対する入力パラメータの詳細

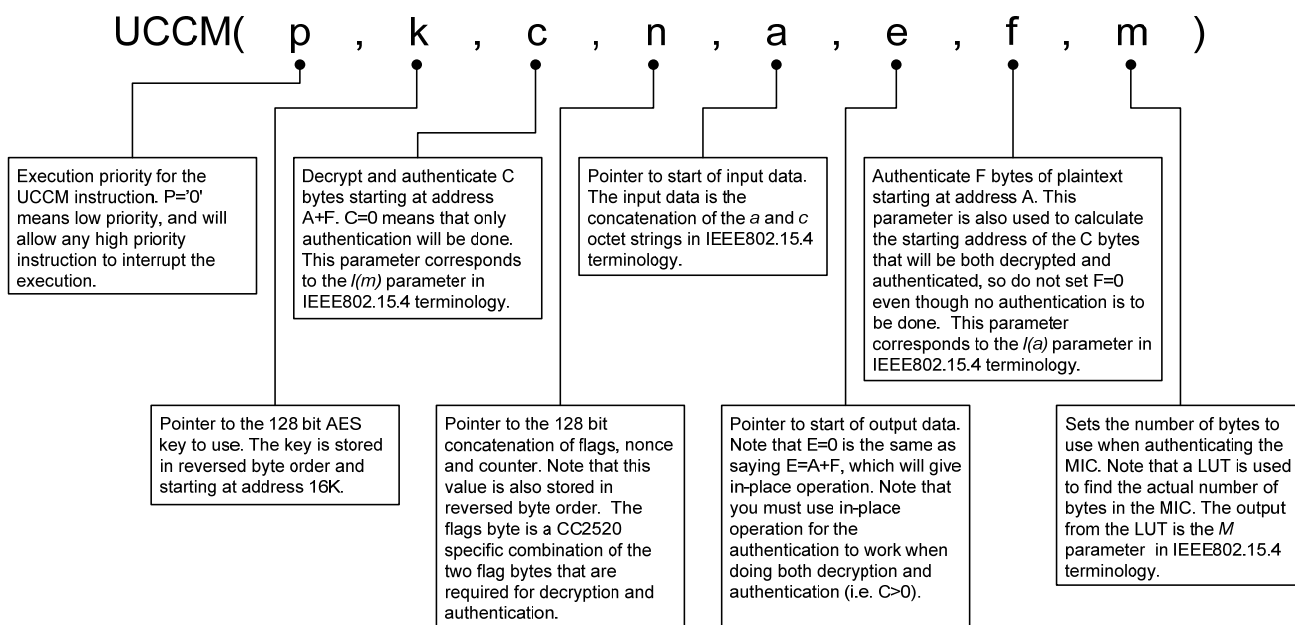


図 39: UCCM 命令に対する入力パラメータの詳細

26.9 IEEE802.15.4-2006 の例

このセクションでは、IEEE802.15.4-2006 [2] の Annex C で示されている CCM* の例を再現するために、ステップバイステップの詳細なガイドを示します。これらの例で使用しているアドレスは、無作為に選択したものです。代わりに他のアドレスを使用することもできます。例で命令に対して指定したすべてのパラメータは 16 進表記を採用し、セクション 13.3 では SPI 通信を行うときに必要とされる正確なビット順序について説明しています。

26.9.1 CCM* を使用する認証のみ

この例では MAC ビーコン・フレームを使用し、CCM/UCCM 命令を使用して認証を適用する方法を示します。

```
//Write frame data to RAM
//Start at address 0x200
MEMWR(A={02 00} D={08 d0 84 21 43 01 00 00 00 00 48 de ac 02 05 00 00 00 00 55 cf 00 00 51 52 53 54})
//Write key to RAM in reverse byte order
//Start at address 0x230
MEMWR(A={02 30} D={cf ce cd cc cb ca c9 c8 c7 c6 c5 c4 c3 c2 c1 c0})
//Write concatenation of flags, nonce and counter to RAM in reversed byte order
//Start at address 0x240
MEMWR(A={02 40} D={00 00 02 05 00 00 00 01 00 00 00 00 48 de ac 09})
//Do CCM operation with high priority
//Append the output to the frame data (by setting the output address E to 0x000)
CCM(P={01} K={23} C={00} N={24} A={200} E={000} F={1a} M={02})
```

CCM 命令から予期される出力は {22 3B C1 EC 84 1A B5 53} です。

レーザで認証コードを検証するために、以下の手順が必要です。

RAM 内でアドレス 0x200 以降にフレーム・データがすでに存在していることを想定しています。

```
//Write key to RAM in reverse byte order
//Start at address 0x230
MEMWR(A={02 30} D={cf ce cd cc cb ca c9 c8 c7 c6 c5 c4 c3 c2 c1 c0})
//Write concatenation of flags, nonce and counter to RAM in reversed byte order
//Start at address 0x240
MEMWR(a={02 40} D={00 00 02 05 00 00 00 01 00 00 00 00 48 de ac 09})
//Do UCCM operation with low priority
UCCM(P={00} K={23} C={00} N={24} A={200} E={2c0} F={1a} M={02})
//Read DPUSTAT register at address 0x02C to check whether the authentication passed or not
REGRD(A={2c})
```

26.9.2 CCM* を使用する暗号化のみ

この例では MAC データ・フレームを使用し、CCM/UCCM 命令を使用してペイロードの暗号化/復号化を適用する方法を示します。

```
//Write frame data to RAM
//Start at address 0x200
MEMWR(A={02 00} D={69 dc 84 21 43 02 00 00 00 00 48 de ac 01 00 00 00 00 48 de ac 04 05 00 00 00 61 62 63 64})
//Write key to RAM in reverse byte order
//Start at address 0x230
MEMWR(A={02 30} D={cf ce cd cc cb ca c9 c8 c7 c6 c5 c4 c3 c2 c1 c0})
//Write concatenation of flags, nonce and counter to RAM in reversed byte order
//Starting at address 0x240
MEMWR(A={02 40} D={00 00 04 05 00 00 00 01 00 00 00 00 48 de ac 01})
//Do CCM instruction.
CCM(P={01} K={23} C={04} N={24} A={200} E={2c0} F={1a} M={00})
```

CCM 命令から予期される出力は {D4 3E 02 2B} です。

レーザでフレームを復号化するために、以下の手順が必要です。RAM 内でアドレス 0x200 以降にパック化データがすでに存在していることを想定しています。

```
//Write key to RAM in reverse byte order
//Start at address 0x230
MEMWR(A={02 30} D={cf ce cd cc cb ca c9 c8 c7 c6 c5 c4 c3 c2 c1 c0})
//Write concatenation of flags, nonce and counter to RAM in reversed byte order
//Starting at address 0x240
MEMWR(A={02 40} D={00 00 04 05 00 00 00 01 00 00 00 00 48 de ac 01})
//Decrypt the frame data and put the plaintext at address 0x2C0.
UCCM(P={01} K={23} C={04} N={24} A={200} E={2c0} F={1a} M={00})
```

CCM 命令から予期される出力は {61 62 63 64} です。

26.9.3 CCM* を使用する暗号化と認証の組み合わせ

この例では MAC コマンド・フレームを使用し、CCM/UCCM 命令を使用して暗号化/復号化と認証の両方を適用する方法を示します。

```
//Write frame data to RAM
//Start at address 0x200
MEMWR(A={02 00} D={2b dc 84 21 43 02 00 00 00 00 48 de ac ff ff 01 00 00 00 00 48 de ac 06 05 00 00
00 01 CE})
//Write key to RAM in reverse byte order
//Start at address 0x230
MEMWR(A={02 30} D={cf ce cd cc cb ca c9 c8 c7 c6 c5 c4 c3 c2 c1 c0})
//Write concatenation of flags, nonce and counter to RAM in reversed byte order
//Starting at address 0x240
MEMWR(A={02 40} D={00 00 06 05 00 00 00 01 00 00 00 00 48 de ac 09})
//Do CCM instruction.
//Replace the last byte in the plaintext frame with the ciphertext and encrypted MIC by setting
the output address E to 0x000.
CCM(P={01} K={23} C={01} N={24} A={200} E={000} F={1d} M={02})
```

CCM 命令から予期される出力は {D8 4F DE 52 90 61 F9 C6 F1} です。

レシーバでフレームを復号化するために、以下の手順が必要です。RAM 内でアドレス 0x200 以降にパック化データがすでに存在していることを想定しています。

```
//Write key to RAM in reverse byte order
//Start at address 0x230
MEMWR(A={02 30} D={cf ce cd cc cb ca c9 c8 c7 c6 c5 c4 c3 c2 c1 c0})
//Write concatenation of flags, nonce and counter to RAM in reversed byte order
//Starting at address 0x240
MEMWR(A={02 40} D={00 00 06 05 00 00 00 01 00 00 00 00 48 de ac 09})
//Decrypt the frame data and authenticate the MIC. Note that the output address E is set to 0x000.
UCCM(P={01} K={23} C={01} N={24} A={200} E={000} F={1d} M={02})
//Read DPUSTAT register at address 0x02C to check whether the authentication passed or not
REGRD(A={2c})
```

UCCM 命令から予期されるプレーンテキスト出力は {CE} です。

27 パケット・スニフing

パケット・スニフingは、CC2520 によって送信または受信されようとしているデータを観測する、侵入を伴わない方法です。パケット・スニフingは、クロックと、クロックの立ち上がりエッジでサンプリングされるデータ信号を出力します。2 個のパケット・スニファ信号は、GPIO 出力として観測できます。正確なタイムスタンプを取得するには、SFD 信号も出力する必要があります。

CC2520 は 250kbps のデータ・レートのみをサポートしているので、パケット・スニファ・クロック周波数は 250kHz です。データは各バイトの MSB を先頭にしてシリアル出力されます。これは実際の RF 出力とは逆ですが、データ処理にとってはこの方が便利です。SPI スレーブを使用してデータ・ストリームを受信することも可能です。

TX モードでフレームをスニフingする場合、変調器によって TX FIFO から読み取られたデータは、パケット・スニファから出力されるデータと同じものです。ただし、自動的な CRC 生成が有効になっている場合、パケット・スニファはこれらの 2 バイトを出力しません。代わりに、パケット・スニファは CRC バイトを 0x8080 という値で置き換えます。(自動的な CRC チェックが有効になっている場合)受信したフレームの最後の 2 バイトとしてこの値が生成されることは決してありません。したがって、スニフingされたデータに対応するレシーバは、CC2520 によって送信されたフレームと、CC2520 によって受信されたフレームを区別することができます。

RX モードでフレームをスニフingする場合、復調器によって RX FIFO に書き込まれたデータは、パケット・スニファから出力されるデータと同じものです。つまり、最後の 2 バイトは受信した CRC の値と、CRC OK/RSSI/相関/SRCRESINDEX の値のどちらかであり、構成の設定によっては後者は CRC の値を自動的に置き換えることとなります。

GPIO ピン上でパケット・スニフing・データを観測するには、MDMTEST1 レジスタ内でパケット・スニファ・モジュールを有効にし、GPIO_n レジスタのいずれかに対して正しい GPIO 構成を書き込む必要があります。

28 レジスタ

以下の表で、CC2520 内の構成レジスタに関するメモリ・マッピングを示します。

FREG レジスタは、REGRD 命令と REGWR 命令を使用してアクセスできます。0x000 ~ 0x01F のアドレス空間にあるレジスタ(灰色の網掛け)は、BSET 命令と BCLR 命令を使用してアクセスすることもできます。

SREG レジスタは、MEMRD 命令と MEMWR 命令のみを使用してアクセスできます。

残りのアドレス範囲の詳細については、図 11: CC2520 メモリ・マップ を参照してください。

表 20: レジスタの概要

アドレス(16進)	+0x000	+0x001	+0x002	+0x003
FREG レジスタ				
0x000	FRMFILT0	FRMFILT1	SRCMATCH	
0x004	SRCSHORTEN0	SRCSHORTEN1	SRCSHORTEN2	
0x008	SRCEXTEN0	SRCEXTEN1	SRCEXTEN2	
0x00C	FRMCTRL0	FRMCTRL1	RXENABLE0	RXENABLE1
0x010	EXCFLAG0	EXCFLAG1	EXCFLAG2	
0x014	EXCMASKA0	EXCMASKA1	EXCMASKA2	
0x018	EXCMASKB0	EXCMASKB1	EXCMASKB2	
0x01C	EXCBINDX0	EXCBINDX1	EXCBINDY0	EXCBINDY1
0x020	GPIOCTRL0	GPIOCTRL1	GPIOCTRL2	GPIOCTRL3
0x024	GPIOCTRL4	GPIOCTRL5	GPIOPOLARITY	
0x028	GPIOCTRL		DPUCON	
0x02C	DPUSAT		FREQCTRL	FREQTUNE
0x030	TXPOWER	TXCTRL	FSMSTAT0	FSMSTAT1
0x034	FIFOPCTRL	FSMCTRL	CCACTRL0	CCACTRL1
0x038	RSSI	RSSISTAT		
0x03C	RXFIRST		RXFIFOCNT	TXFIFOCNT

アドレス(16進)	+0x000	+0x001	+0x002	+0x003
SREG レジスタ				
0x040	CHIPID		VERSION	
0x044	EXTCLOCK		MDMCTRL0	MDMCTRL1
0x048	FREQEST		RXCTRL	
0x04C	FSCTRL		FSCAL0	FSCAL1
0x050	FSCAL2	FSCAL3	AGCCTRL0	AGCCTRL1
0x054	AGCCTRL2	AGCCTRL3	ADCTEST0	ADCTEST1
0x058	ADCTEST2		MDMTEST0	MDMTEST1
0x05C	DACTEST0	DACTEST1	ATEST	DACTEST2
0x060	PTEST0	PTEST1	RESERVED	
0x064-0x077				
0x078			DPUBIST	
0x07C	ACTBIST		RAMBIST	

注: マッピングされていないアドレスにアクセスすると、MEMADDR_ERROR 例外が生成されます。この方法は、0x064 ~ 0x079 のアドレス空間、および 0x07F より上のアドレスに当てはまりません。その他のマッピングされていないアドレス(0x003 など)では、MEMADDR_ERROR は生成されません。

28.1 レジスタ設定の更新

このセクションでは、デフォルト値から更新する必要があるレジスタ設定の概要を説明します。CC2520 が LPM2 になった後、アクティブ・モードに復帰するたびに、毎回これらの値を書き込む必要があることに注意してください。

表 21: デフォルト値から更新する必要があるレジスタ

レジスタ名	アドレス(16進)	新しい値(16進)	概要
TXPOWER	030	32	0dBm の出力電力を設定します。このレジスタ内では、表 17 に示されている値のみを使用します。
CCCTRL0	036	F8	CCA のスレッショールドを約 -108dBm から約 -84dBm の入力レベルに引き上げます。
MDMCTRL0	046	85	sync ワードより前に 2 個のゼロシンボルを必要とすることにより、sync ワードが検出される可能性を低くします。
MDMCTRL1	047	14	SFD で両方のシンボルが相関スレッショールドを上回る相関値を持っている必要があるという要件を排除することにより、sync が検出される可能性を高め、相関スレッショールドを引き上げることにより sync ワードが検出される可能性を低くします。
RXCTRL	04A	3F	RX 関連アナログ・モジュール内の電流を調整します。
FSCTRL	04C	5A	シンセサイザ内の電流を調整します。

レジスタ名	アドレス(16進)	新しい値(16進)	概要
FSCAL1	04F	2B	VCO 内の電流を調整します。
AGCCTRL1	053	11	AGC 制御ループに対応するターゲット値を調整します。
ADCTEST0	056	10	ADC 性能を最適化します。
ADCTEST1	057	0E	ADC 性能を最適化します。
ADCTEST2	058	03	ADC 性能を最適化します。

28.2 レジスタ・アクセス・モード

以下の表の「モード」列は、各ビットに対して許可されるアクセスの種類を示します。表 22 に、さまざまな選択肢の意味を示します。

表 22: レジスタ・ビットのアクセス・モード

モード	概要
R	読み取り
W	書き込み
R0	定数 0 の読み取り
R1	定数 1 の読み取り
W1	1 の書き込みのみが可能
W0	0 の書き込みのみが可能
R*	値の読み取りはレジスタの実際の値ではなく、モジュールから見た値になります。通常、(キャリブレーション、動的制御などによって)構成値が自動的に生成されるか、レジスタの値を手動で上書きする場合に、この方法を使用します。以下に、AGCCTRL2 レジスタに関する構造の例を示します。

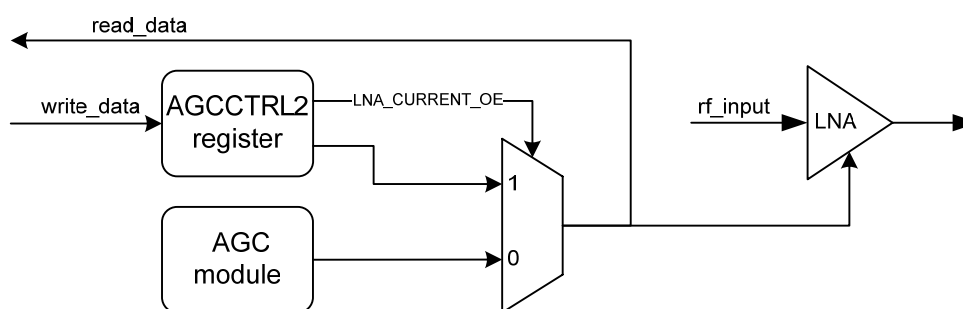


図 40: R* レジスタ・アクセス・モードに関するハードウェア構造の例

28.3 レジスタの説明

各レジスタに対応する見出しは、次のパターンに従って作成したものです。
 <レジスタ名>, A <アドレス>, R <リセット値>, <レジスタの短い説明>

FRMFILTO, A 0x000, R 0x0D, フレーム・フィルタリング

ビット番号	ビットのニーモニック(略称)	リセット値	モード	概要
7	RESERVED	0	R/W	予約済み 常に '0' を書き込みます
6:4	FCF_RESERVED_MASK[2:0]	000	R/W	フレーム制御フィールド(FCF)の予約済み部分をフィルタリングする目的で使用します。 FCF_RESERVED_MASK[2:0] は、FCF[9:7] と AND 処理されます。結果が 0 ではなく、フレーム・フィルタリングが有効になっている場合、フレームは拒否されます。
3:2	MAX_FRAME_VERSION[1:0]	11	R/W	フレーム制御フィールド(FCF)のフレーム・バージョン・フィールドをフィルタリングする目的で使用します。 FCF[13:12](フレーム・バージョン・サブフィールド)が MAX_FRAME_VERSION[1:0] を上回っていて、フレーム・フィルタリングが有効になっている場合、フレームは拒否されます。
1	PAN_COORDINATOR	0	R/W	デバイスが PAN コーディネータである場合、送信先アドレスが指定されていないフレームを受け入れるためにこのビットを "HIGH" に設定する必要があります(802.15.4(b) のセクション 7.5.6.2 の規定に従って)。 0 - デバイスは PAN コーディネータではありません 1 - デバイスは PAN コーディネータです
0	FRAME_FILTER_EN	1	R/W	フレーム・フィルタリングを有効にします。 このビットが設定されている場合、CC2520 は 802.15.4(b) のセクション 7.5.6.2「third filtering level」で規定されたフレーム・フィルタリングを実行します。FRMFILTO[6:1] と FRMFILT1[7:1] をローカル・アドレス情報と組み合わせると、フィルタリング・アルゴリズムの動作が定義されます。 0 - フレーム・フィルタリングをオフにします (FRMFILTO[6:1]、FRMFILT1[7:1]、SRCMATCH[2:0] はいずれも無視されます)。 1 - フレーム・フィルタリングをオンにします。

FRMFILT1, A 0x001, R 0x78, フレーム・フィルタリング

ビット番号	ビットのニーモニック(略称)	リセット値	モード	概要
7	ACCEPT_FT_4TO7_RESERVED	0	R/W	予約済みフレームを受け入れるかどうかを定義します。予約済みフレームでは、フレーム・タイプ = (100, 101, 110, 111) です。 0 - 拒否 1 - 受け入れ
6	ACCEPT_FT_3_MAC_CMD	1	R/W	MAC コマンド・フレームを受け入れるかどうかを定義します。MAC コマンド・フレームでは、フレーム・タイプ = 011 です。 0 - 拒否 1 - 受け入れ
5	ACCEPT_FT_2_ACK	1	R/W	アクノリッジ応答フレームを受け入れるかどうかを定義します。アクノリッジ応答フレームでは、フレーム・タイプ = 010 です。 0 - 拒否 1 - 受け入れ
4	ACCEPT_FT_1_DATA	1	R/W	データ・フレームを受け入れるかどうかを定義します。データ・フレームでは、フレーム・タイプ = 001 です。 0 - 拒否 1 - 受け入れ
3	ACCEPT_FT_0_BEACON	1	R/W	ビーコン・フレームを受け入れるかどうかを定義します。ビーコン・フレームでは、フレーム・タイプ = 000 です。 0 - 拒否 1 - 受け入れ
2:1	MODIFY_FT_FILTER[1:0]	00	R/W	これらのビットは、フレーム・タイプ・フィルタリングを実行する前に、受信したフレームのフレーム・タイプ・フィールドを変更するために使用されます。この変更は、RX FIFO に書き込まれるフレームに影響を及ぼします。 00 : そのままにします 01 : MSB を反転します 10 : MSB を '0' に設定します 11 : MSB を '1' に設定します
0	RESERVED	0	R/W	予約済み 常に '0' を書き込みます

SRCMATCH, A 0x002, R 0x07, 送信先アドレス・マッチングと保留ビット

ビット番号	ビットのニーモニック(略称)	リセット値	モード	概要
7:3	RESERVED[4:0]	0x00	R/W	予約済み 常に '0' を書き込みます
2	PEND_DATAREQ_ONLY	1	R/W	このビットがセットされている場合、AUTOPEND関数は受信フレームが "DATA REQUEST" MAC コマンド・フレームであることも要求します。
1	AUTOPEND	1	R/W	<p>アクノリッジ応答の保留フラグを自動的に有効にします。</p> <p>フレームを受信するときに、以下の条件が成立する場合、(おそらく)返そうとするアクノリッジ応答内の保留フィールドが自動的に設定されます。</p> <ul style="list-style-type: none"> - FRMFILTO.FRAME_FILTER_EN が設定されます。 - SRCMATCH.SRC_MATCH_EN が設定されます。 - SRCMATCH.AUTOPEND が設定されます。 - 受信フレームが現在の SRCMATCH.PEND_DATAREQ_ONLY 設定と一致しています。 - 受信した送信元アドレスが、送信元比較テーブル・エントリのいずれかと一致しています。このテーブルは、SHORT_ADDR_EN と SHORT_PEND_EN の両方、または EXT_ADDR_EN と EXT_PEND_EN の両方によって有効にされます。 <p>注: SHORT_PEND_EN および EXT_PEND_EN の詳細については、メモリ・マップの説明を参照してください。</p>
0	SRC_MATCH_EN	1	R/W	送信元アドレス・マッチングを有効にします (FRMFILTO.FRAME_FILTER_EN = 0 の場合、このビットは「無視」されます)

SRCSHORTENO, A 0x004, R 0x00, 短いアドレス・マッチング

ビット番号	ビットのニーモニック(略称)	リセット値	モード	概要
7:0	SHORT_ADDR_EN[7:0]	0x00	R/W	<p>24 ビット・ワードである SHORT_ADDR_EN のうち 7:0 の部分は、24 個の短いアドレスからなるテーブル・エントリのそれぞれに対する送信元アドレス・マッチングを有効/無効にします。</p> <p>オプションの安全性機能: 送信元アドレス・マッチング・テーブルであるエントリを更新している最中にそのエントリが使用されないことを保証するには、更新を行う間、対応する SHORT_ADDR_EN ビットを 0 に設定します。</p>

SRCSHORTEN1, A 0x005, R 0x00, 短いアドレスのマッチング

ビット番号	ビットのニーモニック(略称)	リセット値	モード	概要
7:0	SHORT_ADDR_EN[15:8]	0x00	R/W	24 ビット・ワードである SHORT_ADDR_EN のうち 15:8 の部分 SRCSHORTEN0 の説明を参照してください。

SRCSHORTEN2, A 0x006, R 0x00, 短いアドレス・マッチング

ビット番号	ビットのニーモニック(略称)	リセット値	モード	概要
7:0	SHORT_ADDR_EN[23:16]	0x00	R/W	24 ビット・ワードである SHORT_ADDR_EN のうち 23:16 の部分 SRCSHORTEN0 の説明を参照してください。

SRCEXTEN0, A 0x008, R 0x00, 拡張アドレス・マッチング

ビット番号	ビットのニーモニック(略称)	リセット値	モード	概要
7:0	EXT_ADDR_EN[7:0]	0x00	R/W RO	<p>24 ビット・ワードである EXT_ADDR_EN のうち 7:0 の部分は、12 個の拡張アドレスからなるテーブル・エントリのそれぞれに対する送信元アドレス・マッチングを有効/無効にします。</p> <p>書き込みアクセス: 拡張アドレスのうち、テーブル・エントリ n ($0 \sim 7$) から EXT_ADDR_EN[2n] へのマッピングを有効にします。すべての EXT_ADDR_EN[2n + 1] ビットは読み取り専用であり、書き込みを行うときはこれらのビットは無視されます。</p> <p>読み取りアクセス: 拡張アドレスのうち、テーブル・エントリ n ($0 \sim 7$) から EXT_ADDR_EN[2n] と EXT_ADDR_EN[2n+1] の両方へのマッピングを有効にします。</p> <p>オプションの安全性機能: 送信元アドレス・マッチング・テーブルであるエントリを更新している最中にそのエントリが使用されないことを保証するには、更新を行う間、対応する EXT_ADDR_EN ビットを 0 に設定します。</p>

SRCEXTEN1, A 0x009, R 0x00, 拡張アドレス・マッチング

ビット番号	ビットのニーモニック(略称)	リセット値	モード	概要
7:0	EXT_ADDR_EN[15:8]	0x00	R/W	24 ビット・ワードである EXT_ADDR_EN のうち 15:8 の部分 SRCEXTENO の説明を参照してください。

SRCEXTEN2, A 0x00A, R 0x00, 拡張アドレス・マッチング

ビット番号	ビットのニーモニック(略称)	リセット値	モード	概要
7:0	EXT_ADDR_EN[23:16]	0x00	R/W	24 ビット・ワードである EXT_ADDR_EN のうち 23:16 の部分 SRCEXTENO の説明を参照してください。

FRMCTRL0, A 0x00C, R 0x40, フレーム処理

ビット番号	ビットのニーモニック(略称)	リセット値	モード	概要
7	APPEND_DATA_MODE	0	R/W	<p>AUTOCRC = 0 の場合: 無視</p> <p>AUTOCRC = 1 の場合:</p> <p>0: RSSI + crc_ok ビット、および 7 ビットの相関値が各受信フレームの最後に追加されます。</p> <p>1: RSSI + crc_ok ビット、および 7 ビットの SRCRESINDEX が各受信フレームの最後に追加されます。詳細については、表 15 を参照してください。</p>
6	AUTOCRC	1	R/W	<p>TX の場合:</p> <p>1: CRC-16 (ITU-T) がハードウェアによって生成され、送信フレームに追加されます。最後の 2 バイトを TXBUF に書き込む必要はありません。</p> <p>0: CRC-16 をフレームに追加しません。フレームの最後の 2 バイトを手動で生成し、TXBUF に書き込む必要があります(これを行わない場合、TX_UNDERFLOW が生成されます)。</p> <p>RX の場合:</p> <p>1: CRC-16 がハードウェアによってチェックされ、RX FIFO で CRC OK ビットを含む 16 ビットのステータス・ワードによって置き換えられます。ステータス・ワードは、APPEND_DATA_MODE を使用して制御できます。</p> <p>0: フレームの最後の 2 バイト(crc-16 フィールド)は RX FIFO に格納されます。CRC チェック(実行する場合は、手動で実施する必要があります)。</p> <p>この設定はアクノリッジ応答 (AUTOACK を含む) には影響しません。</p>
5	AUTOACK	0	R/W	<p>CC2520 がアクノリッジ応答フレームを自動的に送信するかどうかを定義します。autoack が有効になっている場合、アドレス・フィルタリングによって自動的に受け入れられたすべてのフレームは、アクノリッジ応答要求フラグがセット済みで CRC が有効であるときは、受信から 12 シンボル期間後に自動的にアクノリッジ応答されます。</p> <p>0 - Autoack を無効にします</p> <p>1 - Autoack を有効にします</p>
4	ENERGY_SCAN	0	R/W	<p>RSSI レジスタが、最新の信号強度と、エネルギー・スキャンが有効になったとき以来のピーク信号強度のどちらを保持するかを定義します。</p> <p>0 - 最新の信号強度</p> <p>1 - ピーク信号強度</p>

3:2	RX_MODE[1:0]	00	R/W	<p>RX モードを設定します</p> <p>00: 通常の動作、RX FIFO を使用します。 01: 予約済み 10: RX FIFO ループは RX FIFO 内のオーバーフローを無視し、無限に受け入れを行います。 11: シンボル検索を無効にすることを除き、通常の動作と同じです。シンボル検索が不要な場合、RSSI または CCA の測定に関連して使用します。</p>
1:0	TX_MODE[1:0]	00	R/W	<p>TX をテスト・モードに設定します。</p> <p>00: 通常の動作、TX FIFO を送信します。 01: 予約済み。使用しないでください。 10: TX FIFO ルーピングは TX FIFO のアンダーフローを無視し、循環読み取りを行い、無限に送信します。 11: CRC から取得した擬似ランダム・データを送信し、無限に送信を行います。</p>

FRMCTRL1, A 0x00D, R 0x01, フレーム処理

ビット番号	ビットのニーモニック(略称)	リセット値	モード	概要
7:3	RESERVED	0x00	R0	0 を読み取ります
2	PENDING_OR	0	R/W	<p>出力するアクノリッジ応答フレームの保留データ・ビットを常に '1' に設定するか、メイン FSM とアドレス・フィルタリングによって制御するかを定義します。</p> <p>0 - 保留データ・ビットをメイン FSM とアドレス・フィルタリングによって制御します。 1 - 保留データ・ビットを常に '1' にします。</p>
1	IGNORE_TX_UNDERF	0	R/W	<p>TX アンダーフローを無視するかどうかを定義します。</p> <p>0 - 通常の TX 動作。TX アンダーフローが発生した場合、アンダーフローが検出され、TX は中止されます。 1 - TX アンダーフローを無視します。フレーム長フィールドによって指定されたバイト数を送信します。</p>
0	SET_RXENMASK_ON_TX	1	R/W	<p>STXON を実行したときに RXENABLE レジスタ内のビット 14 を設定するか、そのままにするかを定義します。</p> <p>0: RXENABLE に影響を及ぼしません。 1: RXENABLE のビット 14 をセットします。CC2420 に対する上位互換性を維持する場合に使用します。</p>

RXENABLE0, A 0x00E, R 0x00, RX 有効化

ビット番号	ビットのニーモニック(略称)	リセット値	モード	概要
7:0	RXENMASK[7:0]	0x00	R/W	<p>16ビット・ワードである RXENMASKLSB の LSB 部分</p> <p>RXENABLE はレシーバを有効にします。このレジスタが 0 以外の値である場合、送信後およびアクリッジ応答の送信後に、レシーバがアイドル状態である状況で、メイン FSM がレシーバを有効にします。</p> <p>以下のストローブを使用して RXENMASK を変更できます。</p> <p>SRXON: RXENMASK のビット 15 を設定します。</p> <p>STXON: SET_RXENMASK_ON_TX = '1' の場合、RXENMASK のビット 15 を設定します。</p> <p>SRFOFF: RXENMASK 内のすべてのビットをクリアします。</p> <p>以下の命令は RXENMASK に変更を加えます。</p> <p>RXMASKAND : RXENMASK と、命令に対して指定された 16 ビット・パラメータの間でビット単位の AND 演算を実行します</p> <p>RXMASKOR : RXENMASK と、命令に対して指定された 16 ビット・パラメータの間でビット単位の OR 演算を実行します</p> <p>RXENABLE の書き込みと、ビット単位のアクセスが可能です。BSET と BCLR は、RXENMASK の 16 ビットのうち任意のビットを設定またはクリアします。</p> <p>SRXMASKBITSET と SRXMASKBITCLR は、RXENMASK のビット 13 を設定およびクリアします。</p> <p>複数の要素が RXENMASK を同時に変更しようとする可能性があります。複数の要素による RXENMASK への同時アクセスという状況を処理するために、以下のルールが適用されます。</p> <p>- 2 つの要素が競合していない場合(レジスタのうち異なる部分を変更する)、RXENMASK を変更するための両方の要求が処理されます。</p> <p>- データ・バスは他のすべての要素(BSET、BCLR、REGWR、MEMVWR)より高い優先順位を持ちます。</p> <p>RXENMASK を変更するストローブには、以下の順で優先順位が割り当てられます。</p> <ol style="list-style-type: none"> 1 SRFOFF 2 SXTON 3 SRXON 4 RXMASKOR 5 RXMASKAND 6 SRXMASKBITSET 7 SRXMASKBITCLR

RXENABLE1, A 0x00F, R 0x00, RX 有効化

ビット番号	ビットのニーモニック(略称)	リセット値	モード	概要
7:0	RXENMASK[15:8]	0x00	R/W	16 ビット・ワードである RXENMASK の MSB 部分 RXENABLE0 の説明を参照してください。

EXCFLAG0, A 0x010, R 0x00, 例外フラグ

ビット番号	ビットのニーモニック(略称)	リセット値	モード	概要																		
7:0	EXCFLAG[7:0]	0x00	R/W0	<p>例外 7 ~ 0 に対応する例外保留ビット。例外が発生したときは必ず、このビットがハードウェアによってセットされます。ソフトウェアが実行できるのは、このビットをクリアすることのみです。EXCFKLAG に対して 0 の書き込みのみが実行され、EXFLAG のいずれかのビットに「1」を書き込んでも、レジスタは変化しません。</p> <p>1: この例外が発生し、まだクリアされていません。 0: この例外がまだ発生していないか、ソフトウェアによってクリアされました。</p> <table border="1"> <thead> <tr> <th>ビット番号</th> <th>例外</th> </tr> </thead> <tbody> <tr><td>0</td><td>RF_IDLE</td></tr> <tr><td>1</td><td>TX_FRM_DONE</td></tr> <tr><td>2</td><td>TX_ACK_DONE</td></tr> <tr><td>3</td><td>TX_UNDERFLOW</td></tr> <tr><td>4</td><td>TX_OVERFLOW</td></tr> <tr><td>5</td><td>RX_UNDERFLOW</td></tr> <tr><td>6</td><td>RX_OVERFLOW</td></tr> <tr><td>7</td><td>RXENABLE_ZERO</td></tr> </tbody> </table>	ビット番号	例外	0	RF_IDLE	1	TX_FRM_DONE	2	TX_ACK_DONE	3	TX_UNDERFLOW	4	TX_OVERFLOW	5	RX_UNDERFLOW	6	RX_OVERFLOW	7	RXENABLE_ZERO
ビット番号	例外																					
0	RF_IDLE																					
1	TX_FRM_DONE																					
2	TX_ACK_DONE																					
3	TX_UNDERFLOW																					
4	TX_OVERFLOW																					
5	RX_UNDERFLOW																					
6	RX_OVERFLOW																					
7	RXENABLE_ZERO																					

EXCFLAG1, A 0x011, R 0x00, 例外フラグ

ビット番号	ビットのニーモニック(略称)	リセット値	モード	概要																		
7:0	EXCFLAG[15:8]	0x00	R/W0	<p>例外 15 ~ 8 に対応する例外保留ビット。上記の説明を参照してください。</p> <table border="1"> <thead> <tr> <th>ビット番号</th> <th>例外</th> </tr> </thead> <tbody> <tr><td>8</td><td>RX_FRM_DONE</td></tr> <tr><td>9</td><td>RX_FRM_ACCEPTED</td></tr> <tr><td>10</td><td>SRC_MATCH_DONE</td></tr> <tr><td>11</td><td>SRC_MATCH_FOUND</td></tr> <tr><td>12</td><td>FIFOP</td></tr> <tr><td>13</td><td>SFD</td></tr> <tr><td>14</td><td>DPU_DONE_L</td></tr> <tr><td>15</td><td>DPU_DONE_H</td></tr> </tbody> </table>	ビット番号	例外	8	RX_FRM_DONE	9	RX_FRM_ACCEPTED	10	SRC_MATCH_DONE	11	SRC_MATCH_FOUND	12	FIFOP	13	SFD	14	DPU_DONE_L	15	DPU_DONE_H
ビット番号	例外																					
8	RX_FRM_DONE																					
9	RX_FRM_ACCEPTED																					
10	SRC_MATCH_DONE																					
11	SRC_MATCH_FOUND																					
12	FIFOP																					
13	SFD																					
14	DPU_DONE_L																					
15	DPU_DONE_H																					

EXCFLAG2, A 0x012, R 0x00, 例外フラグ

ビット番号	ビットのニーモニック(略称)	リセット値	モード	概要
7:0	EXCFLAG[23:16]	0x00	R/W0	<p>例外 23 ~ 16 に対応する例外保留ビット。上記の説明を参照してください。</p> <p>ビット番号 例外</p> <p>16 MEMADDR_ERROR</p> <p>17 USAGE_ERROR</p> <p>18 OPERAND_ERROR</p> <p>19 SPI_ERROR</p> <p>20 RF_NO_LOCK</p> <p>21 RX_FRM_ABORTED</p> <p>22 RXBUFMOV_TIMEOUT</p> <p>23 UNUSED</p>

EXCMASKA0, A 0x014, R 0x00, チャンネル A に対する例外マスクング

ビット番号	ビットのニーモニック(略称)	リセット値	モード	概要
7:0	EXCMASKA[7:0]	0x00	R/W	<p>24 ビット・ワードである EXCMASKA のうち 7:0 の部分</p> <p>チャンネル A に対する例外のマスク チャンネル A が特定のピンに対する出力構成として選択されている場合、選択した例外が発生したときに、そのピンで例外の表示が生成されます。対になるチャンネルを選択した場合、マスクされた例外が発生したときに例外が表示されます。</p> <p>各ビットで、次のことが適用されます。</p> <p>1: 選択済み</p> <p>0: マスク済み</p>

EXCMASKA1, A 0x015, R 0x00, チャンネル A に対する例外マスクング

ビット番号	ビットのニーモニック(略称)	リセット値	モード	概要
7:0	EXCMASKA[15:8]	0x00	R/W	<p>24 ビット・ワードである EXCMASKA のうち 15:8 の部分</p> <p>EXCMASKA0 の説明を参照してください。</p>

EXCMASKA2, A 0x016, R 0x00, チャンネル A に対する例外マスクング

ビット番号	ビットのニーモニック(略称)	リセット値	モード	概要
7:0	EXCMASKA[23:16]	0x00	R/W	24 ビット・ワードである EXCMASKA のうち 23:16 の部分 EXCMASKA0 の説明を参照してください。

EXCMASKB0, A 0x018, R 0x00, チャンネル B に対する例外マスクング

ビット番号	ビットのニーモニック(略称)	リセット値	モード	概要
7:0	EXCMASKB[7:0]	0x00	R/W	24 ビット・ワードである EXCMASKB のうち 7:0 の部分 チャンネル B に対する例外のマスク チャンネル B が特定のピンに対する出力構成として選択されている場合、選択した例外が発生したときに、そのピンで例外の表示が生成されます。対になるチャンネルを選択した場合、マスクされた例外が発生したときに例外が表示されます。 各ビットで、次のことが適用されます。 1: 選択済み 0: マスク済み

EXCMASKB1, A 0x019, R 0x00, チャンネル B に対する例外マスクング

ビット番号	ビットのニーモニック(略称)	リセット値	モード	概要
7:0	EXCMASKB[15:8]	0x00	R/W	24 ビット・ワードである EXCMASKB のうち 15:8 の部分 EXCMASKB0 の説明を参照してください。

EXCMASKB2, A 0x01A, R 0x00, チャンネル B に対する例外マスクング

ビット番号	ビットのニーモニック(略称)	リセット値	モード	概要
7:0	EXCMASKB[23:16]	0x00	R/W	24 ビット・ワードである EXCMASKB のうち 23:16 の部分 EXCMASKB0 の説明を参照してください。

EXCBINDX0, A 0x01C, R 0x00, チャンネル X に対する例外バインド

ビット番号	ビットのニーモニック(略称)	リセット値	モード	概要
7:4	RESERVED	0x0	R0	0 を読み取ります
3:0	INSTRUCTIONX	0x0	R/W	チャンネル X に対する命令 例外にバインドする命令の番号です。バインド可能な命令全体のリストについては、GPIO の構成に関するマニュアルを参照してください。

EXCBINDX1, A 0x01D, R 0x12, チャンネル X に対する例外バインド

ビット番号	ビットのニーモニック(略称)	リセット値	モード	概要
7	BINDX_EN	0	R/W	チャンネル X に対する例外バインドを有効にするかどうかを定義します。 0 - バインドを無効にします。 1 - バインドを有効にします。EXCEPTIONX によって指定された例外が発生したときは必ず、INSTRUCTIONX によって指定された命令が実行されます。
6:5	RESERVED	00	R0	0 を読み取ります
4:0	EXCEPTIONX	0x12	R/W	チャンネル X に対する例外 命令にバインドする例外の番号です。有効な構成については、例外の概要を参照してください。

EXCBINDY0, A 0x01E, R 0x00, チャンネル Y に対する例外バインド

ビット番号	ビットのニーモニック(略称)	リセット値	モード	概要
7:4	RESERVED	0x0	R0	0 を読み取ります
3:0	INSTRUCTIONY	0x0	R/W	チャンネル Y に対する命令 例外にバインドする命令の番号です。バインド可能な命令全体のリストについては、GPIO の構成に関するマニュアルを参照してください。

EXCBINDY1, A 0x01F, R 0x12, チャネル Y に対する例外バインド

ビット番号	ビットのニーモニック(略称)	リセット値	モード	概要
7	BINDY_EN	0	R/W	チャネル Y に対する例外バインドを有効にするかどうかを定義します。 0: バインドを無効にします。 1: バインドを有効にします。EXCEPTIONY によって指定された例外が発生したときは必ず、INSTRUCTIONY によって指定された命令が実行されます。
6:5	RESERVED	0	R0	0 を読み取ります
4:0	EXCEPTIONY	0x12	R/W	チャネル Y に対する例外 命令にバインドする例外の番号です。有効な構成については、例外の概要を参照してください。

GPIOCTRL0, A 0x020, R 0x00, GPIO0 の制御

ビット番号	ビットのニーモニック(略称)	リセット値	モード	概要
7	IN0	0	R/W	GPIO0 が入力と出力のどちらであるのかを定義します。アナログ・テスト機能を使用する場合、入力として設定する必要があります。 0 - GPIO0 は出力です 1 - GPIO0 は入力です
6:0	CTRL0	0x00	R/W	GPIO0 の構成 出力の場合: マルチプレクサ・セクタ このピンに対する出力として設定できるすべての信号に関する表については、GPIO の説明を参照してください。 入力の場合: 16 個の命令のうちいずれかによって選択された 4 つの LSB が、この GPIO 入力ラインのアクティブ・エッジでトリガされます。0x0F より大きい値は有効ではありません。

GPIOCTRL1, A 0x021, R 0x27, GPIO1 の制御

ビット番号	ビットのニーモニック(略称)	リセット値	モード	概要
7	IN1	0	R/W	GPIO1 が入力と出力のどちらであるのかを定義します。アナログ・テスト機能を使用する場合、入力として設定する必要があります。 0 - GPIO1 は出力です 1 - GPIO1 は入力です
6:0	CTRL1	0x27	R/W	GPIO1 の構成 詳細については、GPIOCTRL0 レジスタを参照してください

GPIOCTRL2, A 0x022, R 0x28, GPIO2 の制御

ビット番号	ビットのニーモニック(略称)	リセット値	モード	概要
7	IN2	0	R/W	GPIO2 が入力と出力のどちらであるのかを定義します。 0 - GPIO2 は出力です 1 - GPIO2 は入力です
6:0	CTRL2	0x28	R/W	GPIO2 の構成 詳細については、GPIOCTRL0 レジスタを参照してください

GPIOCTRL3, A 0x023, R 0x29, GPIO3 の制御

ビット番号	ビットのニーモニック(略称)	リセット値	モード	概要
7	IN3	0	R/W	GPIO3 が入力と出力のどちらであるのかを定義します。 0 - GPIO3 は出力です 1 - GPIO3 は入力です
6:0	CTRL3	0x29	R/W	GPIO3 の構成 詳細については、GPIOCTRL0 レジスタを参照してください

GPIOCTRL4, A 0x024, R 0x2A, GPIO4 の制御

ビット番号	ビットのニーモニック(略称)	リセット値	モード	概要
7	IN4	0	R/W	GPIO4 が入力と出力のどちらであるのかを定義します。 0 - GPIO4 は出力です 1 - GPIO4 は入力です
6:0	CTRL4	0x2A	R/W	GPIO4 の構成 詳細については、GPIOCTRL0 レジスタを参照してください

GPIOCTRL5, A 0x025, R 0x90, GPIO5 の制御

ビット番号	ビットのニーモニック(略称)	リセット値	モード	概要
7	IN5	1	R/W	GPIO5 が入力と出力のどちらであるのかを定義します。 0 - GPIO5 は出力です 1 - GPIO5 は入力です
6:0	CTRL5	0x10	R/W	GPIO5 の構成 詳細については、GPIOCTRL0 レジスタを参照してください

GPIOPOLARITY, A 0x026, R 0x3F, GPIO ピンの極性制御

ビット番号	ビットのニーモニック(略称)	リセット値	モード	概要
7:6	RESERVED	00	R0	0 を読み取ります
5:0	POLARITY	0x3F	R/W	GPIO ピンの出力極性、または入力エッジを選択します。 0 - 負極性。アクティブ "LOW" のレベルを示します。 入力する場合、立ち下がりエッジがアクティブです。 1 - 正極性。アクティブ "HIGH" のレベルを示します。 入力する場合、立ち上がりエッジがアクティブです。

GPIOCTRL, A 0x028, R 0x00, 他の GPIO オプション

ビット番号	ビットのニーモニック(略称)	リセット値	モード	概要
7	SC	0	R/W	パッドに対する追加のドライブ強度を選択します。 0 - 追加ドライブなし 1 - 追加ドライブあり
6	GPIO_ACTRL	0	R/W	GPIO[1:0] のアナログ機能を制御します。このビットを設定すると、GPIO ピン 0 と 1 の両方がアナログ・パッドに設定されます。 0 - アナログ・パッドを無効にします 1 - アナログ・パッドを有効にします
5:0	GPIO_PUE	0x00	R/W	GPIO パッド 0 ~ 5 でプルアップを個別に有効に設定します。 プルアップは、20kΩ ±15% です。 0 - プルアップを無効にします 1 - プルアップを有効にします

DPUCON, A 0x02A, R 0x01, DPU のタイムアウト制御

ビット番号	ビットのニーモニック(略称)	リセット値	モード	概要
7:1	RESERVED	0x00	R0	0 を読み取ります
0	RXTIMEOUT	1	R/W	<p>RXBUFMOV 命令がタイムアウトになるのが、RX FIFO が空になってから 32μs 後と、空になった直後のどちらであるのかを制御します。32μs のタイムアウトを有効にした場合、RXBUFMOV 命令は現在 RX FIFO にあるバイト数より多くのバイト数を対象にして実行することができます。32μs ごとに 1 バイトが受信されるからです。</p> <p>0 - 即座にタイムアウト 1 - 32μs 後にタイムアウト</p>

DPUSTAT, A 0x02C, R 0x00, DPU ステータス・レジスタ

ビット番号	ビットのニーモニック(略称)	リセット値	モード	概要
7:4	RESERVED	0000	R0	0 を読み取ります
3	AUTHSTAT_H	0	R	<p>認証ステータス。高い優先順位です。高い優先順位の UCBCMAC 命令または UCCM 命令が完了したときに更新され、新しい命令が完了するまではその結果を維持します。直前に実行された認証動作の結果を報告します。</p> <p>0: 認証チェックが失敗しました。 1: 認証チェックが成功したか、認証チェックが実行されませんでした。</p>
2	AUTHSTAT_L	0	R	<p>認証ステータス。低い優先順位です。低い優先順位の UCBCMAC 命令または UCCM 命令が完了したときに更新され、新しい命令が完了するまではその結果を維持します。直前に実行された認証動作の結果を報告します。</p> <p>0: 認証チェックが失敗しました。 1: 認証チェックが成功したか、認証チェックが実行されませんでした。</p>
1	DPUH_ACTIVE	0	R	<p>高い優先順位がアクティブ</p> <p>0: 高い優先順位のどの DPU 命令も現在アクティブではありません。 1: 高い優先順位の DPU 命令が現在アクティブです。</p>
0	DPUL_ACTIVE	0	R	<p>低い優先順位がアクティブ</p> <p>0: 低い優先順位のどの DPU 命令も現在アクティブではありません。 1: 低い優先順位の DPU 命令が現在アクティブです。</p>

FREQCTRL, A 0x02E, R 0x0B, RF 周波数の制御

ビット番号	ビットのニーモニック(略称)	リセット値	モード	概要
7	RESERVED	0	R0	0 を読み取ります
6:0	FREQ[6:0]	0x0B (2405 MHz)	R/W	<p>周波数制御ワード。</p> $f_{RF} = f_{LO} = (2394 + FREQ[6:0]) \text{ MHz}$ <p>freq[6:0] の周波数ワードは、2394 に対するオフセット値です。このデバイスは、2394MHz ~ 2507MHz の周波数範囲をサポートしています。freq[6:0] で使用可能な設定値は、0 ~ 113 の連続した範囲のうちいずれかです。これらを外れた設定値(114-127)を使用した場合、周波数は 2507MHz になります。</p> <p>IEEE802.15.4-2006 では、2405MHz ~ 2480MHz の周波数範囲内で、5MHz の間隔で 16 チャンネルを規定しています。チャンネルには、11 ~ 26 の番号が割り当てられています。IEEE802.15.4-2006 準拠システムの場合、有効な設定値は以下の値のみになります。</p> $\text{freq}[6:0] = 11 + 5(\text{チャンネル番号} - 11)$

FREQTUNE, A 0x02F, R 0x0F, クリスタル・オシレータの周波数チューニング

ビット番号	ビットのニーモニック(略称)	リセット値	モード	概要
7:4	RESERVED	0x0	R0	0 を読み取ります
3:0	XOSC32M_TUNE[3:0]	0xF	R/W	<p>クリスタル・オシレータを最適化します</p> <p>デフォルト設定である "1111" は、XOSC を最適化しないままにします。デフォルト値から他の値に変更すると、オシレータに対する追加の静電容量をオンに切り替え、実質的に XOSC の周波数を低くします。したがって、値を大きくすると、周波数は高くなります。</p>

TXPOWER, A 0x030, R 0x06, 出力電力の制御

ビット番号	ビットのニーモニック(略称)	リセット値	モード	概要
7:0	PA_POWER [7:0]	0x06	R/W	<p>PA 電力制御。このレジスタ内で示されている値のみを使用します。</p> <p>注: TX に移行する前に、この値を、表 17 に示されている値のいずれかに更新する必要があります。</p>

FSMSTAT0, A 0x032, R 0x00, 無線ステータス・レジスタ

ビット番号	ビットのニーモニック(略称)	リセット値	モード	概要
7	CAL_DONE	0	R	<p>この信号は、「キャリブレーション完了」信号としては使用できないことに注意してください。キャリブレーション・モジュールを無効にする準備ができたときに(必ずしもキャリブレーションが完了したことを意味するものではありません)、非常に短い期間のみにわたって(32MHz クロックの 1 サイクル)この信号は "HIGH" になり、したがって SPI 経由でレジスタの読み取りをキャプチャするのは困難です。</p> <p>この信号について、データシートで記述するべきではありません。</p> <p>代わりに、CAL_RUNNING の立ち下がリエッジを使用する必要があります。</p>
6	CAL_RUNNING	0	R	<p>周波数同期のキャリブレーション・ステータスです。</p> <p>0 - キャリブレーションは完了したか、まだ開始されていません。</p> <p>1 - キャリブレーションは進行中です。</p>
5:0	FSM_FFCTRL_STATE[5:0]	-	R	FIFO とフレーム制御(FFCTRL)の有限状態機械に関する現在の状態を返します。

FSMSTAT1, A 0x033, R 0x00, 無線ステータス・レジスタ

ビット番号	ビットのニーモニック(略称)	リセット値	モード	概要
7	FIFO	0	R	RX FIFO 内にデータが存在する場合、FIFO は "HIGH" になります。RX FIFO がオーバーフローしている間は "LOW" になります。
6	FIFOP	0	R	フレーム・フィルタリングを通過して RX FIFO に存在しているデータが FIFOP_THR バイトを上回っている場合、FIFOP は "HIGH" に設定されます。 RX FIFO に少なくとも 1 個の完全なフレームが存在している場合、FIFOP は "HIGH" に設定されます。RX FIFO が読み取られ、FIFO 内が FIFOP_THR バイト未満のままになった時点で FIFOP は再び "LOW" に設定されます。 RX FIFO がオーバーフローしている間は FIFOP は "HIGH" になります。
5	SFD	0	R	TX の場合： 0: SFD を含むフレーム全体が送信された場合、または SFD が送信されなかった場合 1: SFD が送信された場合 RX の場合： 0: SFD を含むフレーム全体が受信された場合、または SFD が受信されなかった場合 1: SFD が受信された場合
4	CCA	0	R	チャネルの評価をクリアします。CCA_MODE の設定値によって異なります。詳細については、CCACTRL1 を参照してください。
3	SAMPLED_CCA	0	R	CCA のサンプリングされた値を保持します。SSAMPLECCA ストローブまたは STXONCCA ストローブが発行されたときは、この値が必ず更新されます。
2	LOCK_STATUS	0	R	PLL がロックされている場合は '1'、それ以外の場合は '0' です。
1	TX_ACTIVE	0	R	ステータス信号で、FFCTRL が送信状態のいずれかである場合はアクティブになります。
0	RX_ACTIVE	0	R	ステータス信号で、FFCTRL が受信状態のいずれかである場合はアクティブになります。

FIFOPCTRL, A 0x034, R 0x40, FIFOP のスレッシュホールド

ビット番号	ビットのニーモニック(略称)	リセット値	モード	概要
7	RESERVED	0	R0	0 を読み取ります
6:0	FIFOP_THR[6:0]	0x40	R/W	FIFOP 信号を生成するときに使用されるスレッシュホールドです

FSMCTRL, A 0x035, R 0x01, FSM オプション

ビット番号	ビットのニーモニック(略称)	リセット値	モード	概要
7:2	RESERVED	0x00	R0	0 を読み取ります
1	SLOTTED_ACK	0	R/W	<p>アクノリッジ応答フレームの送信タイミングを制御します</p> <p>0: アクノリッジ応答を要求する受信フレームが終了した後、12 シンボル期間が経過した後にアクノリッジ応答フレームを送信します。</p> <p>1: アクノリッジ応答を要求する受信フレームが終了した後、12 シンボル期間を上回る最初のバックオフ・スロット境界より後にアクノリッジ応答フレームを送信します。</p>
0	RX2RX_TIME_OFF	1	R/W	<p>フレームの受け入れが終了した後、12 シンボルのタイムアウトを使用するかどうかを定義します。</p> <p>0 - タイムアウトなし。</p> <p>1 - 12 シンボル期間のタイムアウト。</p>

CCACTRLO, A 0x036, R 0xE0, CCA のスレッシュホールド

ビット番号	ビットのニーモニック(略称)	リセット値	モード	概要
7:0	CCA_THR[7:0]	0xE0	R/W	<p>空きチャネルの評価に関するスレッシュホールド値で、RSSI との比較に使用される符号付き数値を 2 の補数で表現したものです。</p> <p>単位は 1dB で、76dBm に対するオフセットです。受信信号がこの値を下回っている場合、CCA 信号は "HIGH" になります。CCA 信号は、CCA ピンと FSMSTAT1 レジスタで利用できます。</p> <p>CCA 信号が誤った動作をすることを避けるために、この値を CCA_HYST-128 を下回る大きさに決して設定してはならないことに注意してください。</p> <p>注: リセット値は、およそ $-32 - 76 = -108\text{dBm}$ の入力レベルに変換されますが、これは感度限界を十分下回っています。このことは、CCA 信号が決して空きチャネルを示していないことを意味します。</p> <p>このレジスタを 0xF8 に更新する必要があります。これは、およそ $-8 - 76 = -84\text{dBm}$ の入力レベルに変換されます。</p>

CCACTRL1, A 0x037, R 0x1A, 他の CCA オプション

ビット番号	ビットのニーモニック(略称)	リセット値	モード	概要
7:5	RESERVED	000	R0	0 を読み取ります
4:3	CCA_MODE[1:0]	11	R/W	00 : CCA を必ず「1」に設定します 01 : $RSSI < CCA_THR - CCA_HYST$ の場合は $CCA = 「1」$ 、 $RSSI \geq CCA_THR$ の場合は $CCA = 「0」$ 10 : フレームを受信していない場合は $CCA = 「1」$ 、それ以外の場合は $CCA = 「0」$ 11 : $RSSI < CCA_THR - CCA_HYST$ であり、受信フレームでない場合は $CCA = 「1」$ 、 $RSSI \geq CCA_THR$ であるか、受信フレームである場合は、 $CCA = 0$ 。
2:0	CCA_HYST[2:0]	010	R/W	CCA ヒステリシス・レベルを設定します。符号なしの値を dB 単位で返します。

RSSI, A 0x038, R 0x80, RSSI ステータス・レジスタ

ビット番号	ビットのニーモニック(略称)	リセット値	モード	概要
7:0	RSSI_VAL[7:0]	0x80	R	対数スケールの RSSI 推定値であり、符号付き数値を 2 の補数で表現したものです。 単位は 1dB であり、オフセットは TBD です [外付けコンポーネントを含む RX チェーンの絶対ゲインに基づく値であり、測定する必要があります]。RSSI の値は、8 シンボル期間にわたって平均されます。最初に RSSI_VAL を読み取る前に、RSSI_VALID ステータス・ビットをチェックする必要があります。 リセット値である -128 は、RSSI の値が無効であることも示します。

RSSISTAT, A 0x039, R 0x00, RSSI 有効ステータス・レジスタ

ビット番号	ビットのニーモニック(略称)	リセット値	モード	概要
7:1	RESERVED	0000 00	RO	0 を読み取ります
0	RSSI_VALID	0	R	RSSI の値は有効です。RX モードに入って 8 シンボル期間が経過した後で発生します。

RXFIRST, A 0x03C, R 0x00, RX FIFO 内の最初のバイト

ビット番号	ビットのニーモニック(略称)	リセット値	モード	概要
7:0	DATA[7:0]	0x00	R	RXFIFO の最初のバイト 注: このレジスタを読み取っても FIFO の内容は変化しません。

RXFIFOCNT, A 0x03E, R 0x00, RX FIFO 内のバイト数

ビット番号	ビットのニーモニック(略称)	リセット値	モード	概要
7:0	RXFIFOCNT[7:0]	0x00	R	RX FIFO 内のバイト数です。符号なし整数です。

TXFIFOCNT, A 0x03F, R 0x00, TX FIFO 内のバイト数

ビット番号	ビットのニーモニック(略称)	リセット値	モード	概要
7:0	TXFIFOCNT[7:0]	0x00	R	TX FIFO 内のバイト数です。符号なし整数です。

CHIPID, A 0x040, R 0x84, チップ ID

ビット番号	ビットのニーモニック(略称)	リセット値	モード	概要
7:0	CHIPID[7:0]	0x84	R	チップの ID 番号です。0x84 = CC2520

VERSION, A 0x042, R 0x00, チップのバージョン番号

ビット番号	ビットのニーモニック(略称)	リセット値	モード	概要
7:0	VERSION[7:0]	0x00	R	チップのバージョンです。符号なし整数です。

EXTCLOCK, A 0x044, R 0x20, クロック出力の制御

ビット番号	ビットのニーモニック(略称)	リセット値	モード	概要																																																																																																			
7:6	RESERVED	00	R0	0を読み取ります																																																																																																			
5	EXTCLOCK_EN	1	RW	<p>クロック・ジェネレータ・モジュールから外部クロックへの出力を有効にするかどうかを制御します。選択したピンでクロックを取得するには、GPIOピンを出力として構成し、GPIOCTRLn レジスタのいずれかで「Clock」を選択する必要があることに注意してください。</p> <p>1 - クロックは動作 0 - クロックはオフ</p>																																																																																																			
4:0	EXT_FREQ	0x00	RW	<p>外部クロックの周波数設定です。周波数の変更を実行してもグリッチ・フリーであり、デューティ・サイクルは 50/50 です。つまり、周波数を変更しても、現在のクロック設定の期間全体が終了するまでは影響を及ぼしません。</p> <table border="1"> <thead> <tr> <th>設定値</th> <th>分割係数</th> <th>周波数[MHz]</th> </tr> </thead> <tbody> <tr><td>00000</td><td>32</td><td>1,00</td></tr> <tr><td>00001</td><td>31</td><td>1,03</td></tr> <tr><td>00010</td><td>30</td><td>1,07</td></tr> <tr><td>00011</td><td>29</td><td>1,10</td></tr> <tr><td>00100</td><td>28</td><td>1,14</td></tr> <tr><td>00101</td><td>27</td><td>1,19</td></tr> <tr><td>00110</td><td>26</td><td>1,23</td></tr> <tr><td>00111</td><td>25</td><td>1,28</td></tr> <tr><td>01000</td><td>24</td><td>1,33</td></tr> <tr><td>01001</td><td>23</td><td>1,39</td></tr> <tr><td>01010</td><td>22</td><td>1,45</td></tr> <tr><td>01011</td><td>21</td><td>1,52</td></tr> <tr><td>01100</td><td>20</td><td>1,60</td></tr> <tr><td>01101</td><td>19</td><td>1,68</td></tr> <tr><td>01110</td><td>18</td><td>1,78</td></tr> <tr><td>01111</td><td>17</td><td>1,88</td></tr> <tr><td>10000</td><td>16</td><td>2,00</td></tr> <tr><td>10001</td><td>15</td><td>2,13</td></tr> <tr><td>10010</td><td>14</td><td>2,29</td></tr> <tr><td>10011</td><td>13</td><td>2,46</td></tr> <tr><td>10100</td><td>12</td><td>2,67</td></tr> <tr><td>10101</td><td>11</td><td>2,91</td></tr> <tr><td>10110</td><td>10</td><td>3,20</td></tr> <tr><td>10111</td><td>9</td><td>3,56</td></tr> <tr><td>11000</td><td>8</td><td>4,00</td></tr> <tr><td>11001</td><td>7</td><td>4,57</td></tr> <tr><td>11010</td><td>6</td><td>5,33</td></tr> <tr><td>11011</td><td>5</td><td>6,40</td></tr> <tr><td>11100</td><td>4</td><td>8,00</td></tr> <tr><td>11101</td><td>3</td><td>10,67</td></tr> <tr><td>11110</td><td>2</td><td>16,00</td></tr> <tr><td>11111</td><td>2</td><td>16,00</td></tr> </tbody> </table>	設定値	分割係数	周波数[MHz]	00000	32	1,00	00001	31	1,03	00010	30	1,07	00011	29	1,10	00100	28	1,14	00101	27	1,19	00110	26	1,23	00111	25	1,28	01000	24	1,33	01001	23	1,39	01010	22	1,45	01011	21	1,52	01100	20	1,60	01101	19	1,68	01110	18	1,78	01111	17	1,88	10000	16	2,00	10001	15	2,13	10010	14	2,29	10011	13	2,46	10100	12	2,67	10101	11	2,91	10110	10	3,20	10111	9	3,56	11000	8	4,00	11001	7	4,57	11010	6	5,33	11011	5	6,40	11100	4	8,00	11101	3	10,67	11110	2	16,00	11111	2	16,00
設定値	分割係数	周波数[MHz]																																																																																																					
00000	32	1,00																																																																																																					
00001	31	1,03																																																																																																					
00010	30	1,07																																																																																																					
00011	29	1,10																																																																																																					
00100	28	1,14																																																																																																					
00101	27	1,19																																																																																																					
00110	26	1,23																																																																																																					
00111	25	1,28																																																																																																					
01000	24	1,33																																																																																																					
01001	23	1,39																																																																																																					
01010	22	1,45																																																																																																					
01011	21	1,52																																																																																																					
01100	20	1,60																																																																																																					
01101	19	1,68																																																																																																					
01110	18	1,78																																																																																																					
01111	17	1,88																																																																																																					
10000	16	2,00																																																																																																					
10001	15	2,13																																																																																																					
10010	14	2,29																																																																																																					
10011	13	2,46																																																																																																					
10100	12	2,67																																																																																																					
10101	11	2,91																																																																																																					
10110	10	3,20																																																																																																					
10111	9	3,56																																																																																																					
11000	8	4,00																																																																																																					
11001	7	4,57																																																																																																					
11010	6	5,33																																																																																																					
11011	5	6,40																																																																																																					
11100	4	8,00																																																																																																					
11101	3	10,67																																																																																																					
11110	2	16,00																																																																																																					
11111	2	16,00																																																																																																					

MDMCTRL0, A 0x046, R 0x45, モデムの制御

ビット番号	ビットのニーモニック(略称)	リセット値	モード	概要
7:6	DEM_NUM_ZEROS[1:0]	01	R/W	<p>sync を検索するときに、sync ワードより前に何個の 0 シンボルを検出する必要があるのか設定します。MDMCTRL1 レジスタで設定された関連のスレッシュホールドを上回る設定値を使用する場合に必要なのは 1 個だけであることに注意してください。</p> <p>00 : 予約済み 01 : 1 個のゼロシンボル 10 : 2 個のゼロシンボル 11 : 3 個のゼロシンボル</p> <p>注: RX を試みる前に、この値を "10" に更新する必要があります。テストの結果、リセット値をそのまま使用すると、フレームではないものを誤ってフレームとして受信することが過度に多くなることが明らかになりました。</p>
5	DEMOD_AVG_MODE	0	R/W	<p>周波数オフセット平均化フィルタの動作を定義します。</p> <p>0 - プリアンプルを比較した後、平均レベルをロックします。次のフレームを検索するときに、周波数オフセットのキャリブレーションを再開します。 1 - 平均レベルを継続的に更新します。</p>
4:1	PREAMBLE_LENGTH [3:0]	0010	R/W	<p>TX モードで SFD より先に送信するプリアンプル・バイトの数(2 個の 0 シンボル)を、2 を基準としたステップでエンコードして表現します。リセット値である 2 は、IEEE 802.15.4 に準拠しています。</p> <p>0000 - 2 個の先頭のゼロバイト 0001 - 3 個の先頭のゼロバイト 0010 - 4 個の先頭のゼロバイト ... 1111 - 17 個の先頭のゼロバイト</p>
0	TX_FILTER	1	R/W	<p>使用する TX フィルタの種類を定義します。通常の TX フィルタは IEEE802.15.4 規格によって定義されているとおりです。帯域外の発射を減らすために、追加のフィルタリングを適用することもできます。</p> <p>0 - 通常の TX フィルタリング 1 - 追加のフィルタリングを有効にします</p>

MDMCTRL1, A 0x047, R 0x2E, モデムの制御

ビット番号	ビットのニーモニック(略称)	リセット値	モード	概要
7:6	RESERVED	00	R0	0 を読み取ります
5	CORR_THR_SFD	1	R/W	<p>SFD 検出の要件を定義します。</p> <p>0 - プリアンプルに関連するゼロシンボルのうちいずれかの相関値は相関スレッシュホールドを上回っている必要があります。</p> <p>1 - プリアンプルに関連するゼロシンボルのうちいずれかの相関値、および SFD に含まれる 0 シンボルの両方が、相関スレッシュホールドを上回っている必要があります。</p> <p>注: RX を試みる前に、この値を '0' に変更する必要があります。この結果、適切な感度を維持することと、SFD の誤検出を減らすことの間で最善のトレードオフを達成できます。</p>
4:0	CORR_THR[4:0]	0x0E	R/W	<p>復調器に関する相関スレッシュホールド値で、SFD 検索を行う前に設定する必要があります。</p> <p>スレッシュホールド値は、レシーバが無線から取得したデータに対して同期を行う方法を調整します。スレッシュホールドが過度に小さい場合、ノイズに含まれる小さい sync を検出する可能性が高くなります。スレッシュホールドが過度に大きい場合、感度は低下しますが、ノイズで sync が検出される可能性は低くなります。</p> <p>DEM_NUM_ZEROS と組み合わせる場合、感度を高く保ち、ノイズで検出される synch の数を減らすように最適化することができます。</p> <p>注: RX を試みる前に、この値を 0x14 に変更する必要があります。テストの結果、リセット値をそのまま使用すると、フレームではないものを誤ってフレームとして受信することが過度に多くなることが明らかになりました。</p>

FREEST, A 0x048, R 0x00, 推定された RF 周波数オフセット

ビット番号	ビットのニーモニック(略称)	リセット値	モード	概要
7:0	FREEST[7:0]	0x00	R	<p>符号付き数値を 2 の補数で表現した値です。キャリアとレシーバの LO の間の周波数オフセットに関する推定値です。オフセット周波数は、FREEST×7800Hz です。DEM_AVG_MODE は、この推定値を有効にする時期を制御します。DEM_AVG_MODE = 0 の場合、sync が見つかるまでこの推定値は更新されます。その後、受信フレームが終わるまで、周波数オフセット推定値は固定されます。DEM_AVG_MODE = 1 の場合、復調器が有効になるまでこの推定値は更新されます。</p>

MDMTEST1, A 0x05B, R 0x08, モデム用のレジスタのテスト

ビット番号	ビットのニーモニック(略称)	リセット値	モード	概要
7:4	RESERVED	0000	R0	0 を読み取ります
3	RESERVED	1	R/W	書き込みを行いません。
2	RFC_SNIFF_EN	0	R/W	0 - パケット・スニファ・モジュールを無効にします。 1 - パケット・スニファ・モジュールを有効にします。受信および送信されたデータは GPIO ピン上で観測できます。
1	MODULATION_MODE	0	R/W	RX / TX に関して、2 つの RF 変調モードのいずれかに設定します。 0 - IEEE 802.15.4 準拠モード 1 - 逆位相、IEEE に準拠しません
0	RESERVED	0	R/W	書き込みを行いません。

0x04A ~ 0x07F のアドレス範囲にある以下のレジスタは、パフォーマンス最適化とテストの目的で用意されているものであり、一般的には書き込みを行わないでください。このデータシートで説明したパフォーマンスを実現するために更新する必要があるレジスタを 表 21: デフォルト値から更新する必要があるレジスタ に示します。

RXCTRL, A 0x04A, R 0x29, RX モジュールのテスト/チューニング

FSCTRL, A 0x04C, R 0x55, シンセサイザのテスト/チューニング

FSCALO, A 0x04E, R 0x24, シンセサイザのテスト/チューニング

FSCAL1, A 0x04F, R 0x29, VCO のテスト/チューニング

FSCAL2, A 0x050, R 0x20, VCO のテスト/チューニング

FSCAL3, A 0x051, R 0x2A, VCO のテスト/チューニング

AGCCTRL0, A 0x052, R 0x5F, AGC のテスト/チューニング

AGCCTRL1, A 0x053, R 0x0E, AGC のテスト/チューニング

AGCCTRL2, A 0x054, R 0x00, LNA のテスト/チューニング

AGCCTRL3, A 0x055, R 0x2E, AGC と AAF のテスト/チューニング

ADCTEST0, A 0x056, R 0x66, ADC のテスト/チューニング

ADCTEST1, A 0x057, R 0x0A, ADC のテスト/チューニング

ADCTEST2, A 0x058, R 0x05, ADC のテスト/チューニング

MDMTEST0, A 0x05A, R 0x05, モデムのテスト/最適化

DACTEST0, A 0x05C, R 0x00, DAC のテスト/チューニング

DACTEST1, A 0x05D, R 0x00, DAC のテスト/チューニング

ATEST, A 0x05E, R 0x00, アナログ・テスト・モードの制御

DACTEST2, A 0x05F, R 0x00, DAC のテスト/チューニング

PTEST0, A 0x060, R 0x00, 電力シャットダウン信号のテスト/チューニング

PTEST1, A 0x061, R 0x00, 電力シャットダウン信号のテスト/チューニング

RESERVED, A 0x062, R 0x00, 現時点で未使用

DPUBIST, A 0x07A, R 0x00, DPU ROM のテスト/チューニング

ACTBIST, A 0x07C, R 0x00, ACT ROM のテスト/チューニング

RAMBIST, A 0x07E, R 0x02, RAM のテスト/チューニング

29 データシートの改訂履歴

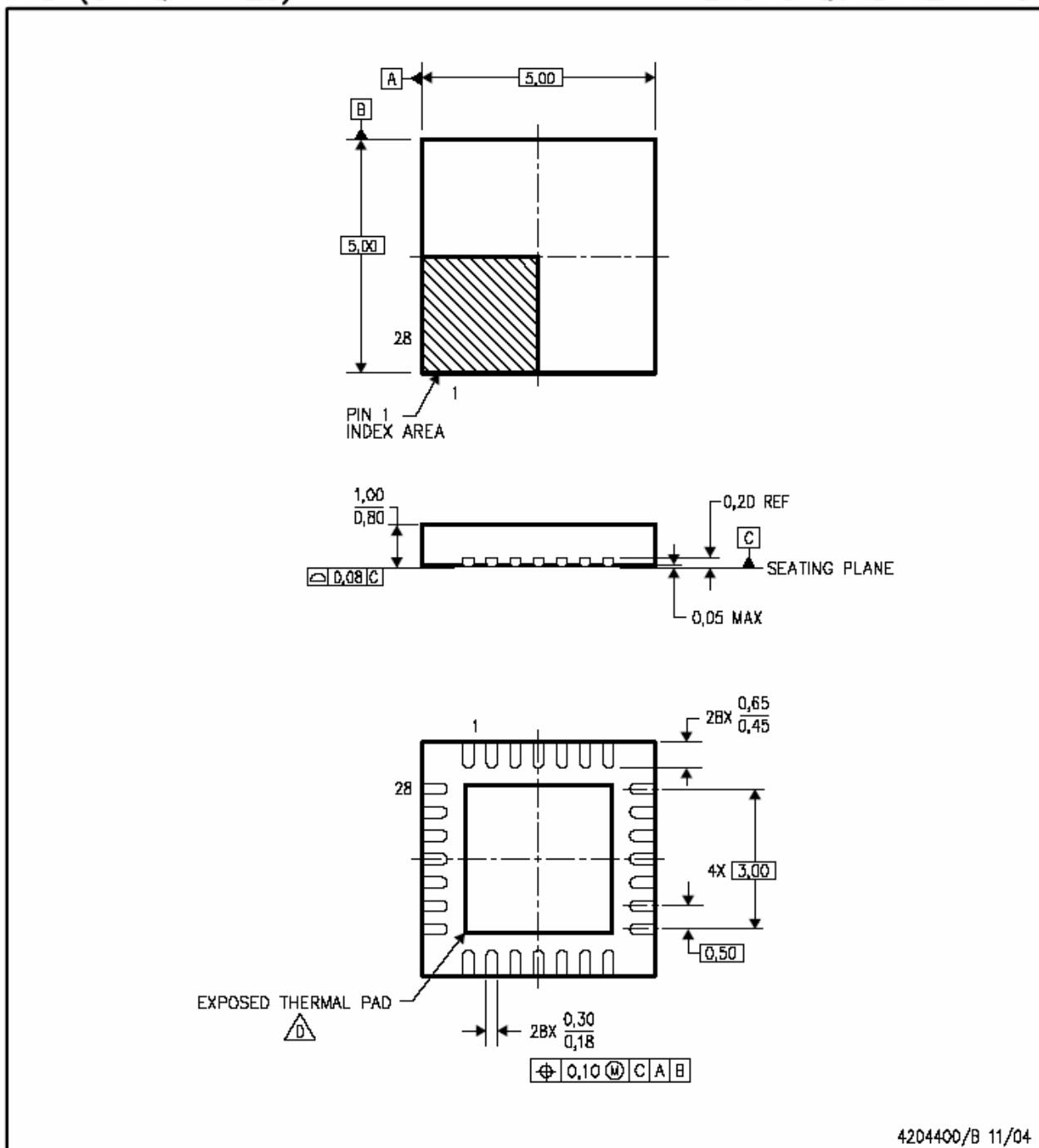
資料番号	リリース日	コメント
SWRS068	2007-12-20	初期リリース

注：ページ番号と図番号は、該当する文書のリビジョンを指しています。

29.1 機械的データ

RHD (S-PQFP-N28)

PLASTIC QUAD FLATPACK



4204400/B 11/04

- 注:
- A. すべての直線寸法は mm 単位です。寸法と公差は ASME Y14-1994 に基づくものです。
 - B. この図面は、予告なく変更される可能性があります。
 - C. QFN(クワッド・フラットパック、リードなし)パッケージの構成。
 - D. このパッケージのサーマル・パッドは、熱性能と機械性能を考慮して、ボードに半田付けされるよう設計されています。
露出したサーマル・パッドの寸法の詳細については、製品データシートを参照してください。
 - E. JEDEC MO-220 に準拠。

ご注意

日本テキサス・インスツルメンツ株式会社(以下TIJといひます)及びTexas Instruments Incorporated(TIJの親会社、以下TIJないしTexas Instruments Incorporatedを総称してTIといひます)は、その製品及びサービスを任意に修正し、改善、改良、その他の変更をし、もしくは製品の製造中止またはサービスの提供を中止する権利を留保します。従ひまして、お客様は、発注される前に、関連する最新の情報を取得して頂き、その情報が現在有効かつ完全なものであるかどうかご確認下さい。全ての製品は、お客様とTIJとの間に取引契約が締結されている場合は、当該契約条件に基づき、また当該取引契約が締結されていない場合は、ご注文の受諾の際に提示されるTIJの標準販売契約約款に従って販売されます。

TIは、そのハードウェア製品が、TIの標準保証条件に従ひ販売時の仕様に対応した性能を有していること、またはお客様とTIJとの間で合意された保証条件に従ひ合意された仕様に対応した性能を有していることを保証します。検査およびその他の品質管理技法は、TIが当該保証を支援するのに必要とみなす範囲で行なわれております。各デバイスの全てのパラメーターに関する固有の検査は、政府がそれ等の実行を義務づけている場合を除き、必ずしも行なわれておりません。

TIは、製品のアプリケーションに関する支援もしくはお客様の製品の設計について責任を負うことはありません。TI製部品を使用しているお客様の製品及びそのアプリケーションについての責任はお客様にあります。TI製部品を使用したお客様の製品及びアプリケーションについて想定される危険を最小のものとするため、適切な設計上および操作上の安全対策は、必ずお客様にてお取り下さい。

TIは、TIの製品もしくはサービスが使用されている組み合わせ、機械装置、もしくは方法に関連しているTIの特許権、著作権、回路配置利用権、その他のTIの知的財産権に基づいて何らかのライセンスを許諾するということは明示的にも黙示的にも保証も表明もしていません。TIが第三者の製品もしくはサービスについて情報を提供することは、TIが当該製品もしくはサービスを使用することについてライセンスを与えるとか、保証もしくは承認をすることを意味しません。そのような情報を使用するには第三者の特許その他の知的財産権に基づき当該第三者からライセンスを得なければならない場合もあり、またTIの特許その他の知的財産権に基づきTIからライセンスを得て頂かなければならない場合もあります。

TIのデータ・ブックもしくはデータ・シートの中にある情報を複製することは、その情報に一切の変更を加えること無く、かつその情報と結び付けられた全ての保証、条件、制限及び通知と共に複製がなされる限りにおいて許されるものとします。当該情報に変更を加えて複製することは不正で誤認を生じさせる行為です。TIは、そのような変更された情報や複製については何の義務も責任も負いません。

TIの製品もしくはサービスについてTIにより示された数値、特性、条件その他のパラメーターと異なる、あるいは、それを超えてなされた説明で当該TI製品もしくはサービスを再販売することは、当該TI製品もしくはサービスに対する全ての明示的保証、及び何らかの黙示的保証を無効にし、かつ不正で誤認を生じさせる行為です。TIは、そのような説明については何の義務も責任もありません。

TIは、TIの製品が、安全でないことが致命的となる用途ないしアプリケーション(例えば、生命維持装置のように、TI製品に不良があった場合に、その不良により相当な確率で死傷等の重篤な事故が発生するようなもの)に使用されることを認めておりません。但し、お客様とTIの双方の権限有る役員が書面でそのような使用について明確に合意した場合は除きます。たとえTIがアプリケーションに関連した情報やサポートを提供したとしても、お客様は、そのようなアプリケーションの安全面及び規制面から見た諸問題を解決するために必要とされる専門的知識及び技術を持ち、かつ、お客様の製品について、またTI製品をそのような安全でないことが致命的となる用途に使用することについて、お客様が全ての法的責任、規制を遵守する責任、及び安全に関する要求事項を満足させる責任を負っていることを認め、かつそのことに同意します。さらに、もし万一、TIの製品がそのような安全でないことが致命的となる用途に使用されたことによって損害が発生し、TIないしその代表者がその損害を賠償した場合は、お客様がTIないしその代表者にその全額の補償をするものとします。

TI製品は、軍事的用途もしくは宇宙航空アプリケーションないし軍事的環境、航空宇宙環境にて使用されるようには設計もされていませんし、使用されることを意図されていません。但し、当該TI製品が、軍需対応グレード品、若しくは「強化プラスチック」製品としてTIが特別に指定した製品である場合は除きます。TIが軍需対応グレード品として指定した製品のみが軍需品の仕様書に合致いたします。お客様は、TIが軍需対応グレード品として指定していない製品を、軍事的用途もしくは軍事的環境下で使用することは、もっぱらお客様の危険負担においてなされるということ、及び、お客様がもっぱら責任をもって、そのような使用に関して必要とされる全ての法的要求事項及び規制上の要求事項を満足させなければならないことを認め、かつ同意します。

TI製品は、自動車用アプリケーションないし自動車の環境において使用されるようには設計されていませんし、また使用されることを意図されていません。但し、TIがISO/TS 16949の要求事項を満たしていると特別に指定したTI製品は除きます。お客様は、お客様が当該TI指定品以外のTI製品を自動車用アプリケーションに使用しても、TIは当該要求事項を満たしていなかったことについて、いかなる責任も負わないことを認め、かつ同意します。

Copyright © 2009, Texas Instruments Incorporated
日本語版 日本テキサス・インスツルメンツ株式会社

弊社半導体製品の取り扱い・保管について

半導体製品は、取り扱い、保管・輸送環境、基板実装条件によっては、お客様での実装前後に破壊/劣化、または故障を起こすことがあります。

弊社半導体製品のお取り扱い、ご使用にあたっては下記の点を遵守して下さい。

1. 静電気

素手で半導体製品単体を触らないこと。どうしても触る必要がある場合は、リストストラップ等で人体からアースをとり、導電性手袋等をして取り扱うこと。

弊社出荷梱包単位(外装から取り出された内装及び個装)又は製品単品で取り扱いを行う場合は、接地された導電性のテーブル上で(導電性マットにアースをとったもの等)、アースをした作業者が行うこと。また、コンテナ等も、導電性のものを使うこと。

マウンタやはんだ付け設備等、半導体の実装に関わる全ての装置類は、静電気の帯電を防止する措置を施すこと。前記のリストストラップ・導電性手袋・テーブル表面及び実装装置類の接地等の静電気帯電防止措置は、常に管理されその機能が確認されていること。

2. 温・湿度環境

温度: 0~40、相対湿度: 40~85%で保管・輸送及び取り扱いを行うこと。(但し、結露しないこと。)

直射日光があたる状態で保管・輸送しないこと。

3. 防湿梱包

防湿梱包品は、開封後は個別推奨保管環境及び期間に従ひ基板実装すること。

4. 機械的衝撃

梱包品(外装、内装、個装)及び製品単品を落下させたり、衝撃を与えないこと。

5. 熱衝撃

はんだ付け時は、最低限260以上の高温状態に、10秒以上さらさないこと。(個別推奨条件がある時はそれに従うこと。)

6. 汚染

はんだ付け性を損なう、又はアルミ配線腐食の原因となるような汚染物質(硫黄、塩素等ハロゲン)のある環境で保管・輸送しないこと。はんだ付け後は十分にフラックスの洗浄を行うこと。(不純物含有率が一定以下に保証された無洗浄タイプのフラックスは除く。)

以上