

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5807508号
(P5807508)

(45) 発行日 平成27年11月10日(2015.11.10)

(24) 登録日 平成27年9月18日(2015.9.18)

(51) Int.Cl.		F I			
H03K	3/02	(2006.01)	H03K	3/02	P
H03K	3/354	(2006.01)	H03K	3/354	C
H03B	5/04	(2006.01)	H03B	5/04	E

請求項の数 5 (全 38 頁)

(21) 出願番号	特願2011-232472 (P2011-232472)	(73) 特許権者	514315159
(22) 出願日	平成23年10月24日(2011.10.24)		株式会社ソシオネクスト
(65) 公開番号	特開2013-90312 (P2013-90312A)		神奈川県横浜市港北区新横浜2丁目10番23
(43) 公開日	平成25年5月13日(2013.5.13)	(74) 代理人	100094525
審査請求日	平成26年7月1日(2014.7.1)		弁理士 土井 健二
		(74) 代理人	100094514
			弁理士 林 恒徳
		(72) 発明者	橋 大
			神奈川県横浜市港北区新横浜二丁目10番23 富士通セミコンダクター株式会社内
		(72) 発明者	有賀 健太
			神奈川県横浜市港北区新横浜二丁目10番23 富士通セミコンダクター株式会社内

最終頁に続く

(54) 【発明の名称】 発振回路を有するマイクロコントローラ

(57) 【特許請求の範囲】

【請求項1】

CPUと、

周波数調整信号に応じて周波数が可変制御されるクロックを生成して前記CPUに供給する発振回路と、

温度を検知する温度センサと、

前記温度センサにより検知される温度が所定温度変動したことに応答して前記CPUにより実行される周波数調整プログラムと、所定温度における前記周波数調整信号と前記発振回路の発振周波数との関係を示す調整信号対周波数関係データと、前記周波数調整信号が所定値の場合における温度と前記発振回路の発振周波数との関係を示す温度対周波数関係データとを格納するメモリとを有し、

少なくとも前記CPUと前記発振回路とがチップ内に構成され、

前記周波数調整プログラムが前記CPUにより実行されることで、前記CPUが、前記温度対周波数関係データに基づいて、前記温度センサにより検知される現在温度と前記所定温度に対する周波数の比率を抽出し、前記調整信号対周波数関係データを前記比率に基づいて修正した修正調整信号対周波数関係データに基づいて、前記発振回路の発振周波数を目標の周波数に制御する周波数調整信号を演算し、

前記演算した周波数調整信号が前記発振回路に設定されるマイクロコントローラ。

【請求項2】

請求項1において、

更に、前記現在温度から前記温度センサにより検知される温度が所定温度変化したことを検知して前記CPUに周波数調整割込信号を出力する温度変化検出ユニットを有し、前記CPUは前記周波数調整割込信号に応答して前記周波数調整プログラムを実行するマイクロコントローラ。

【請求項3】

請求項1または2において、

前記発振回路は、リング状に接続された複数のインバータ回路と、前記リング状の接続経路内に設けられた抵抗及びキャパシタとを有し、前記抵抗の抵抗値が前記周波数調整信号に応じて可変設定されるCR発振回路であり、

前記CR発振回路は、第1の周波数調整信号に応じて前記抵抗の抵抗値が可変設定され、当該可変設定された抵抗値に基づいて第1の発振周波数で発振する低周波CR発振回路と、第2の周波数調整信号に応じて前記抵抗の抵抗値が可変設定され、当該可変設定された抵抗値に基づいて前記第1の発振周波数より高い第2の発振周波数で発振する高周波CR発振回路とを有し、

前記第1の周波数調整信号は、前記周波数調整プログラムを前記CPUが実行することで演算され、前記低周波CR発振回路に設定され、

更に、前記第1の周波数調整信号により調整された前記低周波CR発振回路の第1の発振周波数と、前記高周波CR発振回路の第2の発振周波数との周波数比が、前記目標の周波数に対応する周波数比になるように、前記第2の周波数調整信号を生成し、前記第2のCR発振回路に設定する高周波CR発振制御ユニットを有し、

前記第2の高周波CR発振回路の出力クロックが前記CPUに供給されるマイクロコントローラ。

【請求項4】

請求項1または2において、

前記発振回路は、互いに接続されたインダクタとキャパシタを有し、前記キャパシタの容量値が前記周波数調整信号に応じて可変設定されるLC発振回路であるマイクロコントローラ。

【請求項5】

請求項1または2において、

前記発振回路は、リング状に接続された複数のインバータ回路と、前記リング状の接続経路内に設けられた抵抗及びキャパシタとを有し、前記抵抗の抵抗値が前記周波数調整信号に応じて可変設定されるCR発振回路であるマイクロコントローラ。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、発振回路を有するマイクロコントローラに関する。

【背景技術】

【0002】

マイクロコントローラ(MCU)などの集積回路は、チップ上にCR発振回路あるいはリングオシレータなどを内蔵し、MCUにそのクロック信号を内蔵発振回路から供給する。水晶振動子あるいはセラミック振動子を使用した発振回路では、発振回路の出力周波数が安定するまでの起動時間が大きく、発振周波数精度は低下しても、より起動時間の短いCR発振回路やリングオシレータなどをクロック源として使用することが望ましい場合があるためである。より具体的には、発振回路の停止と起動を頻繁に繰り返すような応用では、発振回路の起動時の待ち時間を短くしてこの期間の消費電力を削減することが、総合的なシステムの性能向上の観点から望ましい。また、コスト削減の目的で、外付けの水晶発振子などではなく、オンチップの発振回路を使用する場合もある。

【0003】

図1は、従来のCR発振回路の一例を示す図である。これは、例えば、特許文献6(特開2009-246793号公報)、特許文献7(特開2011-135349号公報)などに記載されている。図

10

20

30

40

50

1のIV1, IV2, IV3はインバータ回路を, C1, C2は容量を, R1は抵抗を, ND1からND4は発振回路内部のノードを, GNDはGND電位(0V)を示している。

【0004】

図2は, 図1の回路の動作波形の例を示す図である。ND1, ND2, ND3の波形はCMOS回路の一般的な出力波形, 矩形波となる。ND4の波形はND2との容量カップリングによりND2の電位変化の時刻にND4の電位が容量結合によりND2と同じ方向に変化し, その後緩やかにND3の電位と抵抗R1により充放電される波形となる。

【0005】

容量C1, C2は等しい容量であり, インバータIV3によりノードND3が立ち下がるとき, ノードND4は電源電圧VddからR1, C1, C2によるCR時定数により低下し, ノードND4がインバータIV1の閾値電圧Vdd/2を越えると, ノードND2が立ち下がり, 容量C1のカップリングによりノードND4がグランドGNDまで立ち下がる。インバータIV3によりノードND3が立ち上がる時の動作は, 上記と逆の動作になる。したがって, CR時定数の2倍の周期で, 各ノードが発振クロックを出力する。

【0006】

図1のような単純な回路構成で, CR発振回路が実現できる。このCR発振回路では, ノードND4の充放電波形の時定数が抵抗R1と容量C1, C2で決まる。抵抗と容量の値を適切に選ぶことで, 発振周波数を目標の値とすることができる。ただし, 図1の従来回路では, 集積化した容量の容量値, 抵抗の抵抗値の変動により, 発振周波数が変動する問題があった。

【0007】

図3は, 従来発振回路の一例を示す図である。例えば, 特許文献7(特開2011-135349号後方)に示されている。図3のIV1, IV4, IV5はインバータ回路を, C1, C2は容量を, NMn(nは整数)はNMOSトランジスタを, PMn(nは整数)はPMOSトランジスタを, Vddは+の電源(例えば1.8V)を, GNDはGND電位(0V)を, NDn(nは整数)は発振回路内部のノードを, PBはPMOSトランジスタのバイアス電位を, NBはNMOSトランジスタのバイアス電位を, VREG1はVddを発生するレギュレータ回路を, CLK1は出力クロックを, IREFG1は温度依存性を調整可能な基準電流発生回路を, TRT[4:0]は温度依存性調整信号を, IBIASSTRIMは基準電流を, IDAC1は電流DAC回路を, TRF[7:0]は周波数調整のための信号を, IBIASROSCは電流DAC回路の出力電流を示している。図3において, 図1の回路に対応するノード, 素子には同じ符号を与えて, 対応関係が分かるように示した。なお, 図3のインバータ回路IV2, IV4は, 図1のIV2に対応する。

【0008】

図3の回路では, 容量C1の一端(ND5)を一定電圧を電源とするインバータIV2(PM3, NM3)で駆動し, ノードND4の信号振幅を温度によらず一定に制御している。周波数を温度に依存しないように設計するために, トランジスタPM2, NM2に流れる電流を温度に依存せず一定とするように回路を構成する。バイアス電位PB, NBにはトランジスタPM2, NM2に流れる電流が一定となるようにバイアス電位を与える。

【0009】

図3の回路では, PB, NBの電位を発生するバイアス発生回路は, 全てチップ上に集積化され, 温度に依存しない電流を発生するために, 例えば, 以下に説明するような回路構成が採用されている。バンドギャップ回路と同様の原理を利用して, 絶対温度(T)に正に比例(PTAT, Proportional To Absolute Temperatureと一般に称されている)する電流(以下, PTAT電流と呼ぶ), および, 絶対温度(T)に負に比例(Complementary To Absolute Temperature: 絶対温度に対して負の線形依存性)する電流(以下, CTAT電流と呼ぶ), を発生する。これら, PTAT電流とCTAT電流を適切な割合で加算することで, 温度に依存しない(あるいは任意の依存性を持った)基準電流IBIASSTRIMを発生する。さらに, バイアス電流IBIASROSCの絶対値を調整するために, 電流DACを使用し, 基準電流IBIASSTRIMからIBIASROSCを発生する。このバイアス電流がトランジスタPM2, NM2に生成され, ノードND4の充放電時定数の温度依存性を抑制している。その結果, 図3の発振回路

10

20

30

40

50

では、温度依存性が抑制され、電圧振幅Vdd、容量、電流により発振周波数を決定することができる。電流を変化させることで、製造時の容量の絶対値の変動を相殺し、発振周波数を、目標の値に調整することができる。

【 0 0 1 0 】

図4は、従来のCR発振回路の他の一例を示す図である。例えば、特許文献6（特開2009-246793号公報）に示されている。図4のIVn（nは整数）はインバータ回路を、C1、C2は容量を、RV10からRV13は抵抗を、ND1、ND2、ND4、ND30からND33は発振回路内部のノードを、Vddは+の電源（例えば1.8V）を、GNDはGND電位（0V）を、RSEL0からRSEL3は同じ構成の可変抵抗の選択のための回路を、PMD1はPMOSトランジスタを、NMD1はNMOSトランジスタを、NA1は2入力NAND回路を、NO1は2入力NOR回路を、IVD1はインバータ回路を、SEL00からSEL11は可変抵抗選択のための信号を示している。

10

【 0 0 1 1 】

SEL00からSEL11の選択信号のどれか1つをHとして、残りをLとすることで、RSEL0からRSEL3のうち1つが選ばれる。選ばれた回路の出力がLあるいはHに駆動され、残りの非選択の回路の出力はハイインピーダンスとなる。選ばれた回路の出力からノードND4までの間の抵抗の総和が抵抗となるので、図1の回路の抵抗R1を、選択信号SEL00からSEL11で可変制御することができる。

【 0 0 1 2 】

以上説明したように、LSIに集積可能な発振回路としてCR発振回路が知られており、各種の回路が実用に供されている。CR発振回路は集積化が容易である一方、周波数精度に限

20

【 0 0 1 3 】

図5は、PLLのVCOの例を示す図である。例えば、特許文献8（特開2005-176357号公報）に開示されている。図5中、ITAIL1は電流源を、PM01、PM02はPMOSトランジスタを、NM01、NMO2はNMOSトランジスタを、L1はインダクタを、CV1A、CV1B、CV2A、CV2Bはバラクタ（電圧制御可変容量）を、Vddは+の電源を、GNDはGND電位（0V）を、Vadjは周波数調整

30

【 0 0 1 4 】

インダクタL1と容量CV1A、CV1B、CV2A、CV2Bが共振回路を構成し、その共振周波数付近で回路が発振する。これらのLC共振回路は例えばノイズの発生などにより自ら発振する。そして、トランジスタ回路はその共振回路の損失に相当するエネルギーを供給する。

【 0 0 1 5 】

例えば、非特許文献（1）などでも説明されているように、インダクタと容量の値の温度依存性は比較的小さいが、インダクタL1をオンチップに集積すると、インダクタの直列抵抗の影響が無視できない。例えば、アルミなどの配線でインダクタを構成すると、温度上昇に伴って、直列抵抗が上昇する。このため実際のインダクタの直列抵抗が無視できない回路では、例えば、発振周波数は温度の上昇（25 から100 の上昇）にともなって、1%程度発振周波数が小さくなる。

40

【 0 0 1 6 】

図5の回路では、インダクタの直列抵抗の温度依存性に由来する発振周波数の温度依存性を補償するための制御電圧Vauxと、PLLのフィードバックループにおいて発振周波数を制御するための制御電圧Vadjを備えている。発振周波数の温度依存性を制御電圧Vauxを利用して補償し、それとは別に、周波数制御用の電圧Vadjを設けることで、制御電圧Vadjにより制御可能な周波数範囲が、温度変動により上限にばらついてPLLとしての周波数引き込み範囲が狭くなることを防止している。

50

【 0 0 1 7 】

また，特許文献 1 0，非特許文献 2 では，図 5 に類似の周波数制御のためのパラクタと，温度依存性調整のためのパラクタを設けた LC 発振回路が示されている。PLL の VCO ではなく，水晶発振回路や，セラミック発振回路の代替発振回路としての構成例が示されている。

【 先行技術文献 】

【 特許文献 】

【 0 0 1 8 】

【 特許文献 1 】 特開昭 53-060149 号 公 報

【 特許文献 2 】 特開平 4-139911 号 公 報

10

【 特許文献 3 】 特開平 9-83309 号 公 報

【 特許文献 4 】 特開 2002-33644 号 公 報

【 特許文献 5 】 特開 2009-111967 号 公 報

【 特許文献 6 】 特開 2009-246793 号 公 報

【 特許文献 7 】 特開 2011-135349 号 公 報

【 特許文献 8 】 特開 2005-176357 号 公 報

【 特許文献 9 】 特開 2011-518535 号 公 報

【 特許文献 1 0 】 USP 7,872,541

【 非特許文献 】

【 0 0 1 9 】

20

【 非特許文献 1 】 M. S. McCorquodale, S. M. Pernia, J. D. O' Day, G. Carichner, and S. Kubba, "A 9.2 mW 528/66/50 MHz monolithic clock synthesizer for mobile uP platforms," in Proc. IEEE Custom Integrated Circuits Conf. (CICC), 2005, pp. 523-526.

【 非特許文献 2 】 M. S. McCorquodale, J. D. O' Day, S. M. Pernia, G. Carichner, S. Kubba, and R.B. Brown, "A Monolithic and Self-Referenced RF LC Clock Generator Compliant With USB 2.0," IEEE J. Solid-State Circuits, vol. 42, no. 2, pp. 385-399, Feb. 2007.

【 発明の概要 】

【 発明が解決しようとする課題 】

30

【 0 0 2 0 】

しかしながら，図 1 の CR 発振回路では，集積化した容量の製造時の容量値，抵抗の抵抗値の変動により，発振周波数が変動する問題がある。特に温度変化に伴う抵抗値の変動に起因する発振周波数の変動を抑制することが求められる。

【 0 0 2 1 】

図 3 の CR 発振回路では，製造時の容量の値，抵抗の値の変動による周波数の変動を，製造後調整でき，また，周波数の温度による変動も補償できる。しかしながら，バンドギャップを利用した回路が必要であり，温度依存性を調整できるバイアス電流回路や，電流の絶対値を調整するための DAC 回路が必要となり，回路が複雑になる問題がある。また，周波数の温度依存性を正確に補償するためには，回路個体毎に温度を変化させて周波数を測定し，温度依存性を調整する必要があるため，製造後の周波数調整が複雑になる問題もある。

40

【 0 0 2 2 】

さらに，図 4 の回路は，抵抗を等価的にスイッチで切り替えて，周波数を調整することで，複雑なバイアス回路を使用せずに，周波数を調整する特性を実現している。しかしながら，使用している抵抗の温度依存性は補償されないため，使用する抵抗の温度依存性が大きい場合には，発振周波数を高い精度で設定できない。

【 0 0 2 3 】

また，図 5 の LC 発振回路においては，簡単な方法で温度依存性を調整する方法が求められる。

【 0 0 2 4 】

50

そこで、本発明の目的は、複雑なバイアス回路を使用せず、抵抗の温度依存性を補償可能なMCU搭載に適したCR発振回路またはLC発振回路を提供することにある。

【課題を解決するための手段】

【0025】

マイクロコントローラの第1の側面は、CPUと、

周波数調整信号に応じて周波数が可変制御されるクロックを生成して前記CPUに供給する発振回路と、

温度を検知する温度センサと、

前記温度センサにより検知される温度が所定温度変動したことに応答して前記CPUにより実行される周波数調整プログラムと、前記周波数調整信号と前記発振回路の発振周波数との関係を示す調整信号対周波数関係データと、温度と前記発振回路の発振周波数との関係を示す温度対周波数関係データとを格納するメモリとを有し、

前記周波数調整プログラムが前記CPUにより実行されることで、前記CPUが、前記温度対周波数関係データと調整信号対周波数関係データとに基づいて、前記温度センサにより検知される現在温度に応じて、前記発振回路の発振周波数を目標の周波数に制御する周波数調整信号を演算し、

前記演算した周波数調整信号が前記発振回路に設定される。

【発明の効果】

【0026】

第1の側面によれば、発振回路の発振周波数が正確に設定可能になる。

【図面の簡単な説明】

【0027】

【図1】従来のCR発振回路の一例を示す図である。

【図2】図1の回路の動作波形の例を示す図である。

【図3】従来の発振回路の一例を示す図である。

【図4】従来のCR発振回路の他の一例を示す図である。

【図5】PLLのVCOの例を示す図である。

【図6】第1の実施の形態におけるCR発振回路とそれを有するマイクロコントローラ(MCU)の構成図である。

【図7】MCUのプログラムの処理の流れの概略を示すフローチャート図である。

【図8】MCUのプログラムの処理の流れの概略を示すフローチャート図である。

【図9】MCUのプログラムの処理の流れの概略を示すフローチャート図である。

【図10】図6のMCUに適したCR発振回路の具体的な回路例を示す図である。

【図11】図10の粗調用の可変抵抗RV04をより詳細に示す図である。

【図12】図11の内部ノードXU0からXU15を駆動する回路を示す図である。

【図13】図10の微調用の可変抵抗RV03をより詳細に示す図である。

【図14】図10の微調用の可変抵抗RV03をより詳細に示す図である。

【図15】図10から図14の回路のTRDL[7:0]の設定値(10進数)と発振周波数の関係の例を示す図である。

【図16】不揮発のフラッシュメモリに格納する低周波CR発振回路CROSC2の周波数設定信号TRDL[7:0]と発振周波数の関係の一例を示す図である。

【図17】低周波CR発振回路CROSC2の周波数と温度の関係の一例を示す図である。

【図18】図10のCR発振回路の周波数と電源電圧Vddの関係の一例を示す図である。

【図19】図6の温度センサTEMP1の具体的な温度特性の一例を示す図である。

【図20】ウインドウコンパレータが有するフルアダーの回路図である。

【図21】ウインドウコンパレータWCMP1の回路図である。

【図22】図6のカウンタ回路COUNTER1とTRDH制御回路TRDHCNTLからなる高周波CR発振制御ユニットの動作を示す図である。

【図23】4MHzCR発振回路CROSC1の制御信号TRDH[7:0]と周波数の関係の一例を示す図である。

10

20

30

40

50

【図24】第1の実施の形態における低周波の100kHz発振回路の変型例を示す図である。

【図25】第1の実施の形態におけるMCUの変型例を示す図である。

【図26】第2の実施の形態におけるMCUの構成を示す図である。

【図27】バラクタの制御電圧とLC発振回路の発振周波数の関係の一例を示す図である。

【図28】バラクタの電圧が一定の場合の、LC発振回路の周波数と温度の関係の一例を示す図である。

【図29】図26の第2の実施の形態におけるMCUのメインプログラムのフローチャート図である。

【図30】図26の回路の割込発生処理のフローチャート図である。

【図31】図26の回路の割り込みプログラムの処理のフローチャート図である。

10

【図32】第3の実施の形態におけるMCUの構成図である。

【発明を実施するための形態】

【0028】

[第1の実施の形態]

図6は、第1の実施の形態におけるCR発振回路とそれを有するマイクロコントローラ(MCU)の構成図である。

【0029】

図6において、MCU1はマイクロコントローラのチップ全体を、CPU1はCPUを、FLASH1はフラッシュメモリを、RAM1はSRAMを、PORT1は汎用ポートを、ADC1はAD変換回路を、MABは、CPU1、FLASH1、PORT1などをつなぐアドレスバスを、MDBは同様にCPU1、FLASH1、PORT1などをつなぐデータバスを、CKBLK1はクロック制御ブロックを、INTERRUPT1は、温度が規定の範囲を超えたときに発生される周波数調整割込信号を、MCLKはCPUに供給されるクロックを、ROSn(nは整数)は抵抗を、IVXn、IVOSn(nは整数)はインバータを、XTAL1は4MHz発振回路用の水晶振動子を、XTALS1は32768Hz発振回路用の水晶振動子を、CXOn、CXOSn(nは整数)は水晶発振回路の容量を、POSC0、POSC1は4MHz水晶発振回路用のチップ外部への接続端子を、POSCS0、POSCS1は32768Hz水晶発振回路用のチップ外部への接続端子を、XCK1は4MHz水晶発振回路用の発振出力を、XCK2は32768Hz水晶発振回路用の発振出力を示す。

20

【0030】

さらに、CROSC1は4MHz発振出力(クロック)CRCK1を発生する高周波CR発振回路を、CROSC2は100kHz発振出力(クロック)CRCK2を発生する低周波CR発振回路を、COSn(nは整数)は容量を、NAOn(nは整数)はNAND回路を、RVO(nは整数)は可変抵抗を、BUFn(nは整数)はバッファ回路を、TRDH[7:0]は4MHz高周波CR発振回路CROSC1の周波数調整信号を、TRDL[7:0]は100kHz低周波CR発振回路CROSC2の周波数調整信号を、ENCR1はCROSC1のイネーブル信号を、ENCR2はCROSC2のイネーブル信号をそれぞれ示す。

30

【0031】

また、SELn(nは整数)はセレクタを、PLL1はPLL回路を、DIV1はN分周回路を、CKRPLはSEL1の出力クロック信号を、CKPLLはPLL1の出力するクロック信号を、CKSL2はSEL2の出力クロック信号を、CKSL3はSEL3の出力クロック信号を、CKSLOWはSEL5の出力クロック信号をしめす。

40

【0032】

そして、COUNTER1はカウンタ回路を、TRDHCNTLは周波数調整信号TRDH[7:0]を制御する制御回路を、REGTRDLはTRDL[7:0]信号を記憶するレジスタを、THは温度の上限を示す設定値を、TLは温度の下限を示す設定値を、REGTHはTHを記憶するレジスタを、REGTLはTLを記憶するレジスタを、WCMP1は温度Tempが上限THを超えるか、あるいは下限TLを下回ったときに周波数調整割込信号INTERRUPT1を発生するウィンドウコンパレータを、TEMP1は低周波発振回路CROSC2の温度を知るための温度センサを、温度センサTEMP1内のVBEはPN接合の順方向の電圧を、TEMP1内のADCはAD変換回路を、TempはTEMP1でデジタル値に変換された温度を、REGTEMPはTempを記憶するレジスタを、ADRTRDLはREGTRDLのCPU1のアドレス空間内でのアドレスを、ADRTHはREGTHのCPU1のアドレス空間内でのアドレスを、ADRTLはREGTL

50

のCPU1のアドレス空間内でのアドレスを，ADRTEMPはREGTEMPのCPU1のアドレス空間内でのアドレスを，DTTRDLはCPU1とREGTRDLで送受信されるデータを，DTTHはCPU1とREGTHで送受信されるデータを，DTTLはCPU1とREGTLで送受信されるデータを，DTTEMPはRETEMP からCPU1へ送られるデータを示している。

【 0 0 3 3 】

他の図でも，IVn (nは整数など)，RVOn (nは整数など)，などは，特に断らない限り同じ内容を示すものとする。図1，図3，図4などの従来回路の対応する回路素子，ノード等には同じ素子名，ノード名を与えて示している。特に断らない限り，図で対応する素子，ノードには同じ名称を与えて，説明の重複を避けるものとする。

【 0 0 3 4 】

[図6のMCUの概略]

図6のMCUは，CR発振回路として，100kHzの低周波クロックを生成する低周波CR発振回路CROSC2と，4MHzの高周波クロックを生成する高周波CR発振回路CROSC1とを有する。そして，低周波CR発振回路CROSC2の可変抵抗RVO2は，温度センサTEMP1により検出された温度に基づいてCPUが周波数調整プログラムを実行して演算で求めた周波数調整信号TRDLにより，その抵抗値が調整される。この周波数調整により低周波CR発振回路CROSC2の出力クロックCRCK2の周波数は，目標の周波数100kHzに高精度に調整される。CPUが周波数調整信号TRDLを演算できるように，フラッシュメモリFLASH1内には，周波数調整プログラムと，周波数調整信号TRDLと低周波CR発振回路の発振周波数との関係を示す調整信号周波数関係データと，温度と低周波CR発振回路の発振周波数（または温度と低周波CR発振回路内の抵抗値）との関係を示す温度周波数関係データとが格納される。

【 0 0 3 5 】

さらに，カウンタCOUNTER1とTRDH制御回路TRDHCNTLとにより構成される高周波CR発振制御ユニットが，低周波CR発振回路の出力クロックCRCK2を基準クロックにして，高周波CR発振回路CROSC1の出力クロックCRCK1の周波数が，目標の（所望の）周波数である4MHzになるように，周波数調整信号TDRHが調整される。この周波数調整信号TDRHにより，高周波CR発振回路CROSC1の可変抵抗RVO1の抵抗値が設定される。

【 0 0 3 6 】

また，レジスタREGTH,REGTLに周波数調整したときの温度から所定の温度上昇または下降したときの上限温度に対応するデータTHと下限温度に対応するデータTLとを格納することで，ウィンドウコンパレータWCMP1が温度センサの検出温度に対応するデジタル値Tempがこれらの上限温度データTHまたは下限温度データTLを越えたか否かを監視し，越えた場合に周波数調整割込信号INTERRUPT1が発生し，それに応答して，CPUは周波数調整プログラムを実行して，再度，低周波CR発振回路CROSC2の周波数調整信号TRDLを演算で求め，低周波CR発振回路CROSC2の発振周波数を所望値の100kHzに調整する。それに応じて，カウンタとTRDH制御回路が自動的に，高周波CR発振回路CROSC1の発振周波数を調整する周波数調整信号TRDHを変更する。

【 0 0 3 7 】

[図6の詳細]

図6の発振回路を有するMCUの動作を以下詳述する。

【 0 0 3 8 】

図6のCROSC2は，図4の従来回路と同様に構成された100kHz程度の低い周波数で発振する低周波CR発振回路である。図4の従来回路と，図6の回路の違いは，図6の回路では，温度センサTEMP1を設けることにある。また，CROSC2の発振周波数は例えば100kHzと低く，それとは別に，4MHz程度の発振周波数で発振するCR発振回路CROSC1を設けた点に違いがある。図6の低周波CR発振回路CROSC2は，図4のような従来回路と異なり，MCU1に搭載された，CPU（CPU1），フラッシュメモリに代表されるような不揮発性メモリFLASH1，温度センサTEMP1と関連するレジスタREGTRDL，REGTH，REGTL，REGTEMP，コンパレータWCMP1，およびFLASH1に記憶されたプログラム（ソフトウェア）の組み合わせによって，その発振周波数を目標の（例えば100kHz）に保つ発振回路として働く。

10

20

30

40

50

【 0 0 3 9 】

温度センサTEMP1の動作を説明する。温度センサTEMP1は、100kHz発振回路CROSC2の温度を測定し、デジタル値に変換して測定温度に対応するデジタル値Tempを出力する。例えば、PNPトランジスタ（あるいはPN接合）に電流を流し、ベースエミッタ間電圧VBEを得る。このVBEは-2mV/ 程度の温度依存性を持つので、例えば、1.2V一定の基準電圧をAD変換の基準電圧として使用し、ベースエミッタ間電圧VBEをAD変換することで、温度を表すデジタル信号Tempを得る。

【 0 0 4 0 】

ベースエミッタ間電圧VBEからTempを得る以外に、例えばVBEを基準として VBEを変換しても温度を示すデジタルデータが得られる。 VBEは、例えば、異なる電流密度にバイ

10

【 0 0 4 1 】

[発振周波数の調整]

低周波CR発振回路CROSC2は100kHz程度の出力CRCK2を得るためのCR発振回路である。図4の従来回路と同様に、抵抗RV02と容量COS2の時定数で発振周波数が決定され、抵抗RV02の値を、スイッチ切り替えで変更することで、発振周波数を調整可能としている。この発振回路CROSC2は、例えば、イネーブル信号ENCR2をHとすることで発振動作を開始し、8bitの周波数制御信号TRDL[7:0]で、出力クロックCRCK2の周波数が変更される。

【 0 0 4 2 】

高周波CR発振回路CROSC1は、4MHz程度の比較的高い周波数を発生するCR発振回路である。図6の回路では、CROSC2と同様の構成の回路として示している。ENCR1をHとすることで動作を開始し、周波数制御信号TRDH[7:0]で周波数を調整可能なCR発振回路として働く。図6では、一例として抵抗RV01と容量COS1で発振周波数を決定し、抵抗RV01の値を変更可能な回路として示しているが、周波数制御信号TRDH[7:0]で発振周波数を調整可能なCR発振回路であればよい。CROSC1の内部の構成は必ずしも、CROSC2と同様である必要はない。

20

【 0 0 4 3 】

低周波CR発振回路CROSC2の発振周波数を100kHz一定に保つ仕組みを、図7、図8、図9などを参照して説明する。図7、図8、図9はMCUのプログラムの処理の流れの概略を示すフローチャート図である。図7は、メインプログラム、図8は温度センサとウインドウコンパレータWCMP1の動作を示している。図9は割り込みプログラムの処理を示している。

30

【 0 0 4 4 】

図7のメインプログラムは、MCUに電源が供給されてリセットが解除されるとCPUにより実行開始される。メインプログラムの実行開始（図7のスタート）後、図6の温度センサTEMP1で、低周波CR発振回路CROSC2の温度を測定する（STEMP1）。CPUは、温度センサTEMP1を動作させて、温度に対応するデジタル値Tempを取得する。プログラムからは、例えば、アドレスADRTEMPにあるレジスタREGTEMPのデータDTTMPを読み出すことで、温度に対応するデジタル値Tempを知ることができる。図7のステップSTEMP1では、仮にTemp=T1度だったものとしている。

【 0 0 4 5 】

ステップSTEMP2ではフラッシュメモリFLASH1から、例えば温度25 のときの周波数調整信号TRDL[7:0]と発振周波数の関係データを読み出す。フラッシュメモリFLASH1には、例えば図16に示すような周波数調整信号対周波数の関係データが保存されている。

40

【 0 0 4 6 】

図16は、不揮発のフラッシュメモリに格納する低周波CR発振回路CROSC2の周波数設定信号TRDL[7:0]と発振周波数の関係の一例を示す図である。横軸は8bit幅の周波数調整信号TRDL[7:0]を10進数で表示した値を、縦軸は発振出力CRCK2の周波数を示している。図16の関係は、例えば、製造後のある回路個体を、温度25 （電源電圧1.8V）で、周波数制御信号TRDL[7:0]を変更したとこの出力クロックCRCK2の周波数を測定して得たものである。周波数調整信号TRDL[7:0]は、8ビットであり256通りの値をとり得るので、256

50

点の設定と周波数の関係をフラッシュメモリFLASH1に格納することが可能である。ただし、図16では、図中黒丸で示した、32組のデータで周波数調整信号TRDL[7:0]と発振周波数の関係を代表してメモリに格納し、保存していない周波数調整信号の周波数は線形補間で求める。フラッシュメモリFLASH1に格納するデータを増やせば、周波数の表現の精度が向上するが、必要な記憶領域が多くなる。必要な精度が得られる範囲内でモデル化し、フラッシュメモリFLASH1にTRDL[7:0]と発振周波数の関係を保存しておく。

【0047】

図16には、周波数調整信号TRDL[7:0]の上位ビットTRDL[7:4]に対する16個の周波数データと、下位ビットTRDL[3:0]に応じて変化する周波数データのうち下位ビットの最大値に対する16個の周波数データとが黒丸で示されている。このような関係になるのは、後述する低周波CR発振回路の構成上、周波数調整信号TRDLの上位ビットと下位ビットを変化させると、一部重複して周波数が変化するからである。

10

【0048】

ステップSTEPM3ではフラッシュメモリFLASH1から、温度と発振周波数の関係データを読み出す。フラッシュメモリFLASH1には、例えば図17のような関係データが保存されている。

【0049】

図17は、低周波CR発振回路CROSC2の周波数と温度の関係の一例を示す図である。図16の関係では、温度25℃で、TRDL[7:0]が105程度のときに、クロックCRCK2の周波数が100kHzとなっている。図17の関係は、例えば、製造後のある回路個体を、電源電圧1.8V、TRDL[7:0]=105の設定で測定して周波数の結果を示したものである。横軸は温度を、縦軸は発振出力CRCK2の周波数を示している。図17に示されるとおり、低周波CR発振回路CROSC2は、使用している抵抗RV02の温度依存性を反映して、温度上昇とともに、同一の周波数調整信号の設定では、抵抗RV02の抵抗値が減少し、それに伴ってCR時定数が大きくなり、発振周波数が上昇する。出力クロックの周期Tは、抵抗R、容量Cとすると、 $T=2CR(\log 2)$ であることから、抵抗Rが減少すれば発振周波数 $f=1/T$ は上昇する。

20

【0050】

この温度に対する発振周波数の関係データは、温度に対する抵抗値の関係データでも良い。抵抗値Rが判明すれば、上記の発振周波数の式 $f=1/T$ により、発振周波数を求めることができる。したがって、フラッシュメモリには、温度に対する抵抗値の関係データを記憶しても良い。

30

【0051】

この図17の温度と発振周波数の関係データを、不揮発メモリFLASH1に記憶しておき、それをステップSTEPM3で読み出す。記憶しているテーブルのデータの組の数、あるいは、式などの係数は、図17の特性を表現する目的に照らして、精度の点から不足がなければ、さまざまな方法が可能である。

【0052】

図7のSTEPM1、STEPM2、STEPM3までで、現在の発振回路CROSC2の温度、温度が25℃のときの周波数調整信号と周波数の関係、周波数調整信号TRDL[7:0]を固定したときの周波数と温度の関係が得られる。ここでは、例えば現在温度T1が70℃だったとする。

40

【0053】

図17の関係データに示されるとおり、特定のTRDL[7:0]での周波数は25℃のときの値(100kHz)に対して70℃では102kHzと2%大きくなる。したがって、図16の関係データの黒丸で示した点の周波数を全体に2%大きくした値が、温度70℃での、周波数調整信号TRDL[7:0]と周波数の関係であることが分かる。

【0054】

図7のメインプログラムのステップSTEPM4では、測定した温度と、図16と図17の関係データとから低周波CR発振回路CROSC2の出力クロックCRCK2の周波数を目標の値100kHzに保つための周波数調整信号TRDL[7:0]の値を計算で求める。計算はCPUにより、周波数調整プログラムを実行することにより行われる。

50

【 0 0 5 5 】

図 1 6 の周波数制御信号対周波数の関係データの例では、TRDL[7:0] = 96 のとき、91.76 kHz、TRDL[7:0] = 111 のとき、107.1kHz、となる数値例となっている。TRDL[7:0] = 96 は上位ビットTRDL[7:4]=6、下位ビットTRDL[3:0]=0に対応し、TRDL[7:0] = 111はTRDL[7:4]=6、TRDL[3:0]=15に対応する。

【 0 0 5 6 】

そこで、図 1 6 中の調整範囲RA1の一次関数は、次のとおりである。すなわち、下位4bitTRDL[3:0]をTRDLLとして表現すると、TRDL[7:0] = 96からTRDL[7:0] = 111の範囲RA1では、周波数は、周波数frequency = (1023*TRDLL) + 91760 Hzの関数としてモデル化できる。

10

【 0 0 5 7 】

図 1 7 の関係データに示されるように、温度が25 から70 に変化したことで周波数が2%大きくなると、上記の関数を102/100倍して周波数frequency = (1043*TRDLL) + 93595 Hzとしてモデル化できる。

【 0 0 5 8 】

この周波数が100kHzとなるTRDL[3:0] = TRDLLを求めると、TRDLL = 6なので、TRDL[7:0] は96+6=102、つまりTRDL[7:0]=102のときに100kHzが得られる。

【 0 0 5 9 】

図 7 のステップSTEPM4では、CPUは、上記のように、測定した温度と、図 1 6 と図 1 7 の関係データとから、現在温度T1で、低周波RC発振回路CROSC2の周波数を100kHzに保つ設定TRDL[7:0]を計算する。

20

【 0 0 6 0 】

そして、ステップSTEPM5では、CPUは、上記の演算で求めた周波数調整信号TRDL[7:0]=102をレジスタREGTRDLに書き込む。それにより、低周波RC発振回路CROSC2の可変抵抗RV02はレジスタREGTRDLから供給される周波数調整信号TRDL[7:0]でその抵抗値を制御され、低周波RC発振回路CROSC2の周波数は目標の100kHzとなる。

【 0 0 6 1 】

また、現在温度T1に対して、例えば10 以上温度が低くなったとき、あるいは、現在温度T1に対して10 以上温度が高くなったとき、再度TRDL[7:0]の値を再計算するために、割り込み信号を発生できるよう、T1-10 をレジスタREGTLに、T1+10 をレジスタREGTHにそれぞれ書き込む。例えば、T1が70 で、再設定しない温度の範囲を+-10 とすれば、THは80、TLは60 となる。

30

【 0 0 6 2 】

そして、低周波CR発振回路RCOSC2の発振周波数に基づいて、カウンタCOUNTER1とTRDH制御回路が、両発振回路CROSC1,CROSC2の出力クロックCRCK1,CRCK2の周波数比が、クロックCRCK1の周波数が目標の周波数4MHzに対する周波数比になるように、高周波側の周波数調整信号TRDHが生成される。よって、CPUによらずに自動的に高周波CR発振回路CROSC1の発振周波数が目標の周波数4MHzに調整される。

【 0 0 6 3 】

低周波側の出力クロックCRCK2が、100kHz一定に制御されているので、カウンタCOUNTER1とTRDH制御回路が、このクロックCRCK2を基準にして、高周波側のクロックCRCK1の周波数を調整することで、クロックCRCK1の周波数を目標の値4MHzに制御する。すなわち、カウンタ回路COUNTER1が、クロックCRCK2の1周期の間、クロックCRCK1の立ち上がりエッジの数を数えることで、両クロックCRCK1、CRCK2の周波数の比を検出する。このカウンタの結果をもとに、制御回路TRDHCNTLが、周波数比が4 0 倍になるようにクロックCROSC1の周波数調整信号TRDH[7:0]を発生し、CRCK1の周波数を4MHzに制御する。カウンタ回路の構成の例は、図 2 2 で再度示すが、100kHzに制御された低周波クロックCRCK2に基づいて、それより高い周波数の4MHzの高周波クロックCRCK1を発生することが可能である。

40

【 0 0 6 4 】

以上のとおり、図 7 の初期化用のメインプログラムを実行した後に、MCUはユーザーブ

50

プログラムの実行を開始する。両発振回路CROSC1, CROSC2を使用する場合には, 温度センサにより温度を監視しながら, 低周波CR発振回路CROSC2の周波数を一定に調整する。図7の初期化時のプログラムの実行により初期化時の周波数調整が行われる。

【0065】

[周波数調整割込]

図8は, 温度センサTEMP1とウインドウコンパレータWCMP1の動作のフローチャート図である。前述のとおり, MCUが発振回路CROSC1, CROSC2を使用する場合には, 温度センサTEMP1を動作させて, CROSC2の温度を監視する。図7の初期化プログラムで説明したT1が70, THは80, TLは60の例で説明すると, 温度センサTEMP1の温度計測結果Tempが60から, 80の範囲内にあるときは, ステップSTEP1とSTEP2を繰り返す。STEP1で温度を測り, STEP2で温度が60から80までの範囲内にあるかを判定する。この判定は, ウインドウコンパレータWCMP1が行う。温度がTL=60より低いか, TH=80より高い場合, ステップSTEP3に実行が移り, ウインドウコンパレータが周波数調整割込信号INTERRUPT1を発生する。そして, CPUの割り込みコントローラが割込信号INTERRUPT1に応答して, 図7のメインプログラムから, 図9の割り込みプログラムに実行を移す。

10

【0066】

図9は, 周波数調整割込プログラムでの処理のフローチャート図である。ステップSTEP11では, 仮に割込時の温度がTemp=TX度だったものとする。周波数割込信号INTERRUPT1が発生しているので, 温度TempはTL=60より低いか, TH=80より高い温度となっている。ここではTemp=TX=90として説明をすすめる。CPUは, アドレスADRTEMPのレジスタREGTEMPを読み出すことで, 測定温度を読み出す。

20

【0067】

ステップSTEP12では, CPUは, フラッシュメモリFLASH1から, 図16に示される温度25のときの周波数調整信号TRDL[7:0]と発振周波数の関係データを読み出す。さらに, CPUは, ステップSTEP13でフラッシュメモリFLASH1から, 図17の温度と発振周波数の関係データを読み出す。

【0068】

ステップSTEP14で, CPUは, 測定温度Tempと図16, 図17の関係データとから, 低周波CR発振回路CROSC2の出力クロックCRCK2を100kHzに保つための周波数調整信号TRDL[7:0]の値を計算で求める。この計算は図7のメインプログラムと同様である。すなわち, 現在温度TXが90だったとすると, 図17の関係データより, 特定のTRDL[7:0]での周波数は25のときの値(100kHz)に対して90では2.5%大きくなる(102.5kHz)。図16の黒丸で示した点の周波数を全体に2.5%大きくした値が, 温度90での周波数調整信号TRDL[7:0]と周波数の関係になる。

30

【0069】

図7の計算と同様の方法で, CPUは, 現在温度TX=90で, 低周波CR発振回路CROSC2の周波数を100kHzに保つ設定TRDL[7:0]を計算する。

【0070】

そして, ステップSTEP15では, CPUが計算で求めたTRDL[7:0]をレジスタREGTRDLに書き込む。発振回路CROSC2はレジスタREGTRDLから供給される周波数制御信号TRDL[7:0]により周波数を制御されるので, 発振回路CROSC2の周波数は目標の値の100kHzとなる。

40

【0071】

温度が90に上昇したことで, 高周波CR発振回路CROSC1の周波数も変動している。しかし, カウンタCOUNTER1とTDH制御回路TRDHCNTLにより, 周波数調整信号TRDHが自動的に生成されるので, 高周波CR発振回路CROSC1の発振周波数も4MHzに調整される。

【0072】

また, 現在温度TX=90に対して, 例えば10以上温度が低くなったとき, あるいは10以上温度が高くなったとき, 再度TRDL[7:0]の値を再計算するために, CPUは, TX-10をレジスタREGTLに, TX+10をレジスタREGTHに書き込む。例えば, TXが90, 再設定しない温度の範囲を+-10とすれば, THは100, TLは80となる。これにより, 温度が10

50

以上変化した時は、再度割込信号が発生し、上記と同様の周波数調整が実行される。

【 0 0 7 3 】

CPUは、図 9 の割込プログラムによるTRDL[7:0]の再計算と、割り込み信号発生の変更を実行したら、プログラムの実行をメインプログラムにもどす。

【 0 0 7 4 】

以上の通り、図 7、図 8、図 9 の処理と、図 6 のハードウェアを使用して、低周波CR発振回路CROSC2の温度を監視し、その温度がある範囲を超えたときに、周波数設定信号を再度計算し、発振回路の出力クロックCRCK2が100kHzにもっとも近くなるように制御することが可能となる。

【 0 0 7 5 】

図 6 のMCUは、高周波CR発振回路CROSC1と低周波CR発振回路CROSC2とを有し、温度計TEMP1による測定温度と、CPUによる周波数調整信号TRDLの計算とにより、低周波CR発振回路CROSC2の出力クロックCRCK2の周波数を100kHz一定に制御している。さらにそのCROSC2の100kHzの出力を基準にして、簡単な回路により、高周波CR発振回路CROSC1の出力クロックCRCK1を4MHzに調整する。

【 0 0 7 6 】

このような構成により以下に説明するような作用効果が得られる。

【 0 0 7 7 】

CR発振回路は、容量と抵抗でCR時定数が決まり、それにより発振周波数が決まる。このようなCR発振回路回路では、発振周波数が低いほうが、発振周波数を目標の値に制御しやすい。なぜなら、発振周期を決める要因には、容量と抵抗で定まるCR時定数以外に、信号をトランジスタ増幅回路で増幅する遅延時間や、インバータ回路の遅延時間が含まれる。ただし、これらの遅延時間は、製造バラツキなどに起因するものであり、そのバラツキの程度はCR時定数に依存せず一定である。したがって、これらトランジスタ回路に起因する遅延時間が、周波数を決定するCR時定数に対する割合が小さくなる周波数、つまり低い周波数で、回路を設計しておく、その周波数は、抵抗と容量で決まる理想的な周波数に近づいていく。

【 0 0 7 8 】

つまり、周波数調整信号TRDLで設定されるCR時定数（低周波故比較的大きい）に対して、遅延時間のバラツキの割合は小さいので、周波数調整信号TRDLによる周波数調整により、高精度に発振周波数を調整できる。CR発振回路の周波数が高いほど、CR時定数は小さくなり、それにたいする遅延時間のバラツキの割合は大きくなるので、高周波CR発振回路を周波数調整信号で調整すると、調整された発振周波数の精度は低くなる。

【 0 0 7 9 】

さらに、製造後の回路個体差が小さくなるので、例えば、発振周波数の温度依存性をより予測しやすくなる。図 3 の従来回路のように、電流の温度依存性を調節する機能を回路に実装し、4MHz程度の高い周波数で発振する回路の周波数と温度の関係を制御することも、原理的には可能である。しかし、高周波にすれば、回路個体間で発振周波数の温度変化の特性が異なる程度が大きくなる。このため発振周波数を広い温度範囲で一定に保とうとすると、製造後の回路毎に発振周波数の温度変化を測定して、その変化が最小となるように回路定数を調整することが必要となる。

【 0 0 8 0 】

一方、図 6 のように、100kHz程度の発振回路の発振周波数は、低周波であるので、抵抗と容量でよく表現でき、トランジスタ特性の影響、アナログ回路で問題となる回路個体毎の特性の違いの影響は相対的に小さくなる。低周波RC発振回路CROSC2の発振周波数の温度依存性に支配的に影響するのは、使用している抵抗素子の温度依存性であり、この温度依存性を高い精度で予測できれば、回路個体毎に温度を変化させて発振周波数を直接測定し、回路定数を調整する必要はなくなる。

【 0 0 8 1 】

例えば、図 3 の従来回路では、基準電流の温度依存性を調整する設定信号、周波数の絶

10

20

30

40

50

対値を調整する信号を必要としたが、図6の本実施の形態の回路では、温度センサでVBEの値をデジタル値に変換することは必要だが、バイアス電流、あるいは電圧などと温度の関係をアナログ回路で調整する回路要素は持っていない。温度と周波数の関係を調整する機能はCPUのプログラムが担っているからである。温度と周波数の関係を調整する機能をプログラムにまかせることでアナログ回路としては、回路が単純化できる利点が得られる。

【0082】

図6の本実施の形態の回路では、回路個体毎の周波数の温度変化が小さくなるよう100kHz程度の低い周波数のCR発振回路CROSC2を選び、このCROSC2の発振周波数を、CPUと周波数調整プログラムと温度計を利用して、100kHz一定に制御し、より高い周波数のCR発振回路CROSC1の周波数を、100kHzを基準に調整している。

10

【0083】

より高い周波数精度が得られる100kHz発振回路(CROSC2)を基準に、4MHz発振回路の周波数を調整することで、100kHz発振回路で得られるのと同じ程度の周波数精度が4MHzのCR発振回路でも得られる効果が得られる。

【0084】

図4の従来回路と同様に、図6の低周波CR発振回路CROSC2を、抵抗をスイッチで切り替えて可変抵抗回路とし、発振周波数を調整する回路とすることで、図3の従来回路で必要だったアナログ回路で構成する基準電流回路、電流DAC回路などが不要となる利点が得られる。

20

【0085】

また、温度に対応する値を温度センサTEMP1でデジタル値Tempに変換すること、不揮発性メモリ(フラッシュメモリFLASH1)に、ある温度での低速CR発振回路CROSC2の発振周波数と周波数調整信号との関係データ(図16)と、抵抗素子の温度と抵抗値の関係データ(図17)とを記憶しておくことで、CPUによる周波数調整プログラムの実行により、低周波CR発振回路の周波数を一定に保つことができる。

【0086】

さらに、最初に周波数設定信号を設定した温度から、ある範囲を超えて温度が変化したときだけ周波数調整割込信号を発生する。これによって、温度変化が緩やかな場合には、CPUは、別のプログラムを実行することができ、周波数調整のための演算時間を抑制することができる。

30

【0087】

[外付け水晶発振子による水晶発振回路]

図6のMCUは、低周波水晶発振回路用の水晶発振子XTALS1を外付けするための外部端子POSC0,1と、高周波水晶発振回路用の水晶発振子CTAL1を外付けするための外部端子POSC0,1とを有する。それぞれの水晶発振子がチップに外付けで設けられると、それらの水晶発振子と、チップ内の抵抗ROS2,ROS1とインバータIVX2,IVX1と、外付けの容量CXOS0,1とCXO0,1とにより低周波水晶発振回路と、高周波発振回路が構成される。これらの発振回路は、水晶発振子を利用しているので高精度の周波数で発振する。

【0088】

40

クロック制御ブロックCKBLK1は、水晶発振子XTAL1を使用する水晶発振回路と、水晶発振子XTALS1を使用する水晶発振回路と、高周波CR発振回路CROSC1と、低周波CR発振回路CROSC2の出力クロックを利用してCPU1のクロック信号MCLKを供給するクロック信号制御ブロックとして機能する。

【0089】

セレクトSEL1はXTAL1を使用する水晶発振回路の出力XCK1と、4MHz程度の周波数で発振するCR発振回路CROSC1の発振出力CRCK1のどちらかを選択して、PLL回路に入力クロックCKRPLとして供給する。セレクトSEL1の制御はプログラムで行われる。

【0090】

チップ外部の水晶振動子XTAL1と容量CX00, CX01を、チップ外部端子POSC0, POSC1に接

50

続することで、MCUチップ内部におかれたインバータIVX1とバイアス用の抵抗ROS1が、例えば、4MHzで発振する水晶発振回路として動作する。図6では、破線で示されたMCU1はチップ内部の構成要素だけを含んでいる。それに対して、一点鎖線のクロックブロックCKBLK1は、チップ外部の部品も含めて示している。

【0091】

前述のとおり、高周波CR発振回路CROSC1の周波数は、低周波CR発振回路CROSC2の出力を基準として、4MHzに調整される。これらのクロックCRCK1とXCK1をPLLの入力信号として選択可能とするのがSEL1である。MCUのユーザーが外付け水晶発振子XTAL1を使用して、水晶発振回路を使用する場合には、その出力クロックの精度が高いため、メインクロックを発生する目的で、高周波CR発振回路CROSC1を使用しないほうが好ましい。セクタSEL1により、PLLの基準クロックをXCK1とCRCK1から選択できるようにすることで、ユーザーによって、XTAL1を使用する場合と使用しない場合とを選択可能にしている。

10

【0092】

PLL1は、セクタSEL1の出力クロックCKRPLから所望の倍率の周波数のクロックCKPLLを生成する。図6のクロック制御ブロックCKBLK1は、このPLLで発生したクロックCKPLLをCPUへのメインクロックMCLKとして供給可能な回路となっている。

【0093】

クロックMCLKはPLLで逡倍されたクロックだけでなく、低い周波数を使用する場合には、低周波のクロックを選択する必要もある。図6のクロック制御ブロックCKBLK1は、そのような低周波クロックとして、100kHz発振回路CROSC2の出力クロックCRCK2と、リアルタイムクロックRTCとして使用されることの多い32768Hz発振回路の出力クロックXCK2とを選択可能となっている。

20

【0094】

チップ外部の水晶振動子XTALS1と容量CXOS0、CXOS1を、チップ外部端子POSCS0、POSCS1に接続することで、MCUチップ内部におかれたインバータIVX2とバイアス用の抵抗ROS2が、例えば、4MHzで発振する水晶発振回路として働く。

【0095】

セクタSEL2は、PLLを使用しない場合に、クロックXCK1、CRCK1、CRCK2、XCK2のいずれかを選んでメインクロックMCLKとして選択する。これにより、水晶発振子XTAL1を使用するユーザー、XTAL1を使用しないユーザー、XTALS1を使用するユーザー、XTALS2を使用しないユーザーがそれぞれ、必要なクロックを、PLLを経由せずにCPU1へ供給できるようになっている。

30

【0096】

PLLを使用する場合のクロックCKPLLとPLLを使用しない場合のクロックCKSL2をセクタSEL3で選択することができる。これらセクタはプログラムでレジスタ設定することで、選択可能になる。セクタSEL3で選ばれたクロックCKSL3をそのままCPU1に供給するか、分周回路DIV1を経由してCPU1に供給するかを、セクタSEL4で選択可能になっている。つまり、分周器DIV1でN分周するか、CKSL3をそのまま使用するかを選択可能とする。

【0097】

クロック制御ブロックCKBLK1のPLLの基準クロックの入力の選択、分周回路、分周数の選択などは、一般的なMCUのクロック制御部分と同じ機能となっている。

40

【0098】

次に、セクタSEL5の目的について説明する。既に説明したように、低周波CR発振回路CROSC2の出力クロックCRCK2をもとに、高周波クロックCRCK1の周波数を調整すれば、その高周波クロックCRCK1の精度は、低周波クロックCRCK2で得られた精度に近くなり、直接、高周波CR発振回路CROSC1の周波数を一定に制御する場合より回路が簡単で、且つ構成になる。32768Hz発振回路を使用しないユーザーにとっては、100kHzCR発振回路CRCK2の出力クロックを基準に高周波CR発振回路CROSC1のクロックの周波数を調整することが必要である。一方、リアルタイムクロックRTCを使用していることで、水晶発振回路の出力クロックXCK2が利用できる場合には、その出力クロックXCK2の精度のほうがクロックCRCK2より高い

50

ので、32768Hzのクロック信号XCK2を基準に高周波クロックCRCK1の周波数を目標の周波数に調整する。セレクタSEL5を設けることで、両クロックXCK2とCRCK2を選択可能になり、ユーザーがCRCK1を調整するための基準クロックをプログラムで選択可能となる。

【 0 0 9 9 】

[MCUの全体構成]

次に、MCUとしての全体の構成を以下、簡単に説明する。マイクロコントローラMCU1は、クロック制御ブロックCKBLK1や、CPU（CPU1として図示した）、フラッシュメモリFLASH1に加えて、一般的なMCUのリソースメモリRAM1や、AD変換回路ADC1、汎用ポートPORT1などから構成される。図6のMCUの構成要素は、一例として示したものであり、さまざまな変形が可能である。

10

【 0 1 0 0 】

図6では、CPUのアドレス空間であるアドレスが割り振られるFLASH1やRAM1、PORT1、ADC1などと同じく、プログラムからアクセス可能なアドレス、ADRTRDL、ADRTL、ADRTL、ADRTL、ADRTLにバスMABを介してアクセスし、レジスタREGTRDL、REGTH、REGTL、REGTEMPの内容をデータDTTRDL、DTTH、DTTL、DTTEMPとして読み書きすることができる。

【 0 1 0 1 】

また、割り込み信号INTERRUPT1をウインドウコンパレータWCMP1が発生し、その温度範囲TH、TLと現在温度Tempをプログラムから知ることができ、100kHz発振回路の周波数設定TRDL[7:0]をプログラムから設定できるレジスタREGTRDLが用意されている回路であればよい。

20

【 0 1 0 2 】

[各回路の構成と動作]

[CR発振回路]

図10は、図6のMCUに適したCR発振回路の具体的な回路例を示す図である。図10において、 IV_n (n は整数)はインバータ回路を、 C_1 、 C_2 は容量を、 ND_1 、 ND_2 、 ND_4 、 ND_7 は発振回路内部のノードを、 V_{dd} は+の電源(例えば1.8V)を、 GND はGND電位(0V)を、 PMA_1 はPMOSトランジスタを、 NMA_1 はNMOSトランジスタを、 NAO_3 は2入力NAND回路を、 RVO_3 、 RVO_4 は可変抵抗を、 $TRDL[3:0]$ 、 $TRDL[7:4]$ は周波数調整信号 $TRDL[7:0]$ の下位ビットと上位ビットを、 EN はイネーブル信号を、 VRC_1 は可変抵抗回路全体を、 BUF_n (n は整数など)はバッファ回路を示す。

30

【 0 1 0 3 】

図10のCR発振回路は、図6内の低周波CR発振回路の例である。ただし、図6内の高周波CR発振回路も同様の回路で構成される。この低周波CR発振回路の可変抵抗回路 VRC_1 は、バッファ BUF_7 と2つの可変抵抗 RVO_3 、 RVO_4 とを有する。これらの可変抵抗は、周波数調整信号 $TRDL[3:0]$ 、 $TRDL[7:4]$ によりそれぞれ可変調整される。また、図10(B)には、図10(A)の可変抵抗回路 VRC_1 が、後述する図11~14によりどのように構成されるかが示されている。

【 0 1 0 4 】

図10(A)のように、可変抵抗回路を可変抵抗 RVO_3 と RVO_4 を階層的に調整することで、図16に示したような周波数調整信号と周波数の関係を得ることができる。また、可変抵抗を RVO_3 と RVO_4 のように階層化することで、回路規模を抑えて、周波数調整の分解能を高くすることができる。可変抵抗 RVO_4 は周波数調整信号の上位4ビット $TRDL[7:4]$ で制御される粗調用可変抵抗であり、可変抵抗 RVO_3 は周波数調整信号の下位4ビット $TRDL[3:0]$ で制御される微調用可変抵抗である。

40

【 0 1 0 5 】

図11は、図10の粗調用の可変抵抗 RVO_4 をより詳細に示す図である。図11のノード ND_4 は図10のノード ND_4 を、抵抗群 RVO_{40} から RVO_{415} は図10の抵抗 RVO_4 を、ノード XU_0 から XU_{15} は周波数調整信号 $TRDL[7:4]$ で選ばれる内部ノードを示す。したがって、ノード ND_4 と内部ノード XU_0 - XU_{15} のいずれかとの間の直列抵抗が、可変抵抗 RVO_4 の抵抗に設定される。 RVO_{40} から RVO_{415} に添えられた数値140kから560kは、 RVO_{40} から RVO_{415} の抵抗の数値例を

50

示している。つまりRV040は、例えば140キロオーム、RV041は、130キロオーム、RV0415は560キロオームに設計する例を示している。このような抵抗値にすることで、図16に示される周波数調整信号TRDL[7:4]により選択される黒点の特性を有する。

【0106】

図12で説明するとおり、上位の周波数調整信号TRDL[7:4]の値により、XU0からXU15のノードの1つが選ばれて、残りのノードはハイインピーダンスとなる。例えば、TRDL[7:4]が1111の場合にはXU15が選ばれて、XU0からXU14は、ハイインピーダンスとなる。それによりND4に接続される抵抗は、XU15とND4の間の抵抗560キロオーム(RV0415)だけとなる。また、例えば、TRDL[7:4]が0000の場合にはXU0が選ばれて、XU1からXU14は、ハイインピーダンスとなる。それによりND4に接続される抵抗は、XU0とND4の間の抵抗560キロオーム(RV0415)に加え、RV0414からRV040すべてを加算した抵抗となる。

10

【0107】

図12は、図11の内部ノードXU0からXU15を駆動する回路を示す図である。図12において、図11の内部ノードXU0-XU15はTRDL[7:4]で選ばれ、ND7は図10のND7を、TRDL[7:4]はRV04の上位の周波数調整信号を、DEC41はTRDL[7:4]をデコードするデコーダ回路を、SELU[15:0]はひとつだけHとなって選択されるTRDL[7:4]をデコードした信号を、RSELU15からRSELU0は同じ構成であり内部ノードXU15からXU0を駆動する回路を、NS、PSはRSELU15からRSELU0のドライバトランジスタNMOS、ドライバトランジスタPMOSの共通のソース電位を、PMD2はPMOSトランジスタを、NMD2はNMOSトランジスタを、NA1は2入力NAND回路を、NO1は2入力NOR回路を、IVD1はインバータ回路を示す。

20

【0108】

図12の抵抗選択ユニットRSELU15からRSELU0は、上位4ビットの周波数調整信号TRDL[7:4]をデコードして内部ノードXU15からXU0のうちのひとつの信号だけをHあるいはLに制御し、残りはハイインピーダンスとする。これにより、ND4に接続される抵抗の値をTRDL[7:4]で調整できる。図10(A)では、ノードND7とND4は同相の信号となっている。

【0109】

図11、図12に示されるとおり、ノードND7とND4は同相で、かつ抵抗を可変できる回路となっている。例えば選択信号SELU15がHのとき、NANDゲートNA1の入力がHなので、ノードND7が変化すると、RSELU15のNANDゲートNA1の出力はノードND7の反転信号となる。同様に、SELU15がHのときのみ、RSELU15のNORゲートNO1の入力がLとなるので、NORゲートNO1の出力はND7の反転信号となる。従って、トランジスタPMD2、NMD2の共通ドレインの出力ノードXU15は、ノードND7の非反転信号となり、RSELU15はバッファ回路BUF7と等価になる。

30

【0110】

逆に、選択信号SELU15がLのときは、NANDゲートNA1の入力がLなので、ND7に関係なく、NANDゲートNA1の出力はHとなる。同様に、SELU15がLのとき、RSELU15のNORゲートNO1の入力がHとなるので、NORゲートNO1の出力はLとなる。従って、RSELU15のPMD2とNMD2がND7に関係なくOFFなので、その出力XU15はハイインピーダンスになる。以上のとおり、選択状態のRSELU0-RSELU15はバッファBUF7となり、非選択状態のRSELU0-RSELU15の出力はハイインピーダンスとなる。

40

【0111】

図13と図14は、図10の微調用の可変抵抗RV03をより詳細に示す図である。図13、図14と図11、図12との関係は、図10(B)に示されるとおりである。

【0112】

図11、図12は、ソースノードPS、NSに接続されるバッファBUF7の出力とノードND4の間の粗調用の可変抵抗RV04を上位4ビットの周波数調整信号TRDL[7:4]で切り替える回路である。それに対して、図13、14では、ソースノードNSとGND、PSとVddの間に設けられた微調用の可変抵抗RV03を、下位4ビットの周波数調整信号TRDL[3:0]で切り替える回路である。

【0113】

50

図 1 3 の抵抗RV03P0からRV03P14はPMOS側の可変抵抗RV03を，図 1 4 の抵抗RV03N0からRV03N14はNMOS側の可変抵抗RV03をそれぞれ示す。また，PMV0からPMV15はPMOSトランジスタを，NMV0からNMV15はNMOSトランジスタを，SELL0からSELL15は4ビットの下位周波数調整信号TRDL[3:0]をデコードした信号を，IVP0からIVP15はインバータ回路を，Vddは+の電源（例えば1.8V）を，GNDはGND電位（0V）を，PS，NSは図 1 2 のNS，PSにつながるノードを，DEC42はTRDL[3:0]をデコードするデコーダを示す。

【 0 1 1 4 】

抵抗RV03P0からRV03P14，抵抗RV03N0からRV03N14に添えられた数値10kは，抵抗の数値例を示している。つまり，例えば，それぞれの抵抗は10キロオームに設計される。

【 0 1 1 5 】

下位の周波数調整信号TRDL[3:0]をデコードすることで，選択信号SELL0からSELL15のどれか1つが選択されてHとなる。図 1 3 では選択された選択信号SELL0-SELL15に対応するPMOSトランジスタだけがONとなり，VddとPSの間の抵抗を0オームから150キロオームまで10キロオームステップで変更できる。図 1 4 でも同様に，選択された選択信号SELL0-SELL15に対応するNMOSトランジスタが1つだけONとなるので，GNDとNSの間の抵抗を，0オームから150キロオームまで10キロオームステップで変更できる。

【 0 1 1 6 】

つまり，下位周波数調整信号TRDL[3:0]で選択された信号SELL0-SELL15に対応するPMOSトランジスタPMV0-PMV15，NMOSトランジスタNM0-NM15がオンになり，微調用可変抵抗RV03の抵抗値が設定される。バッファBUF7が出力を立ち上げる場合は，PMOSトランジスタ側の抵抗値が微調用可変抵抗になり，逆に立ち下げる場合は，NMOSトランジスタ側の抵抗値が微調用可変抵抗になる。

【 0 1 1 7 】

以上のように，図 1 1 から図 1 4 の回路で，図 1 0 のような微調用と粗調用の可変抵抗RV03，RV04を実現できる。

【 0 1 1 8 】

図 1 5 は，図 1 0 から図 1 4 の回路のTRDL[7:0]の設定値（10進数）と発振周波数の関係の例を示す図である。図 1 5 の横軸は，8bit幅の周波数調整信号TRDL[7:0]を10進数で表示した値を，縦軸は周波数を示している。図 1 0 の回路の発振周波数は，容量と抵抗の絶対値で決まるので，製造された回路の抵抗と容量の値に依存して発振周波数は変化する。例えば，電源電圧1.8V，温度25℃一定で，周波数とTRDL[7:0]の関係は容量と抵抗の値で変化する。

【 0 1 1 9 】

図 1 5 の一番上の線は，FAST側にばらついて容量Cと抵抗Rがともに小さくなった場合の周波数と周波数調整信号の関係を示している。この場合CR時定数が小さくなるので周波数が大きくなる。図 1 5 のまんなかの線は，容量Cと抵抗Rがともにtypicalの場合の周波数と周波数調整信号の関係を示している。図 1 5 の一番下の線は，SLOW側にばらついて容量Cと抵抗Rがともに大きくなった場合の周波数と周波数調整信号の関係を示している。この場合，CR時定数が大きくなるので周波数が小さくなる。これらの周波数対調整信号の関係は，温度変動によってそれぞれ上下する。

【 0 1 2 0 】

プロセスばらつきと温度変動分とを考慮して，CR時定数が最も小さく周波数が最も大きくなる場合において最小周波数が目標の100kHzより小さくなり，また，CR時定数が最も大きく周波数が最も小さくなる場合において最大周波数が目標周波数の100kHzより大きくなるのが望まれる。そのように設定できれば，どのようにばらついても，周波数調整信号の設定により，発振周波数を目標周波数の100kHzに調整することが可能となる。

【 0 1 2 1 】

また，図 1 5 において，C,Rが小さくなる場合の目標周波数100kHz近傍の調整信号による周波数調整幅と，C,Rが大きくなる場合の調整幅とが同等になるように，周波数調整信号TRD[7:0]が大きくなるほど，周波数の変更ステップが大きくなるような特性になってい

10

20

30

40

50

る。この周波数特性は、図 1 1 , 1 3 , 1 4 の抵抗値の設定により実現できる。

【 0 1 2 2 】

図 1 5 は、電源電圧1.8V、温度25 のときの、製造ばらつきに依存するCRの値をパラメータとした周波数調整信号TRDL[7:0]と周波数の関係を示している。したがって、製造した回路個体のある周波数調整信号に対する発振周波数を測定することで、いずれの関係になっているかが判明するので、判明した関係がフラッシュメモリに格納される。

【 0 1 2 3 】

例えば、抵抗R、容量Cともに中心値で製造された場合は、そのTRDL[7:0]と周波数の関係は、Typicalの関係になり、図 1 6 のようになる。製造後、出荷前の試験で、TRDL[7:0]と周波数の関係を確認することで、フラッシュメモリに、図 1 6 のような周波数とTRDL[7:0]の関係のデータを記憶する。

10

【 0 1 2 4 】

図 1 6 は、周波数制御信号TRDLと発振周波数との関係を示す図である。前述のとおり、この関係は、既に製造ばらつきの影響は除かれている。そして、図 1 7 に示されるように、温度に依存して抵抗値が変動し発振周波数が変動する。温度が上昇する抵抗値が低下し、発振周波数が上昇する。つまり、図 1 6 の周波数制御信号対発振周波数の関係は、温度の変動に応じて、上下に変動する。

【 0 1 2 5 】

図 1 8 は、図 1 0 のCR発振回路の周波数と電源電圧Vddの関係の一例を示す図である。横軸は電源電圧Vddを、縦軸は周波数を表す。この例は、抵抗と容量はtypical、温度は25、周波数調整信号TRDL[7:0]は105の場合の電源電圧と発振周波数の関係を示している。

20

【 0 1 2 6 】

電源電圧が1.6V程度から2V程度まででは、発振周波数は電源電圧の上昇とともに緩やかに大きくなる。これは、信号を増幅する部分や、インバータなどの遅延時間が電源電圧で変動し、電源電圧が大きいほうが、それら固定遅延が小さくなるためである。回路の発振周波数は主にCR時定数で決定され、CR時定数は、電源電圧に依存しないので、電源電圧が、1.6V程度から2V程度まで変化しても、発振周波数の変化は+-1%程度にとどまる。

【 0 1 2 7 】

但し、さらに電源電圧を下げていくと周波数は大きく低下するが、このような電源電圧の領域を避けて使用し、また、発振回路CROSC2に供給する電源電圧をレギュレータ回路等で1.6-2.0Vの範囲に制御することで、電源電圧の変動に起因する周波数の変動は小さくできる。

30

【 0 1 2 8 】

[温度センサとウインドウコンパレータ]

図 6 において、温度センサTEMP1はPN接合の電圧VBEをADCで変換してデジタル温度測定値としてTempを出力する。そして、ウインドウコンパレータWCOMP1が、TempがレジスタREGTH,REGTLに記録されているの最大温度に対応するTHと最小温度に対応するTLを超えて変化したか否かを検出し、超えた場合に周波数調整割込信号INTERRUPT1を出力する。

【 0 1 2 9 】

図 1 9 は、図 6 の温度センサTEMP1の具体的な温度特性の一例を示す図である。図 1 9 の横軸は温度、縦軸は電圧VBEを示している。PNPトランジスタのエミッタベース間のPN接合に一定電流を流すと、その順方向電圧VBEは、(接合面積、バイアス電流に依存するが)概略、-2mV/ 程度の温度依存性を示し、300Kでの電圧は、例えば600mV程度となることが知られている。

40

【 0 1 3 0 】

図 1 9 は、PNPトランジスタを一定電流でバイアスし、そのVBEが300Kで600mV、VBEの温度依存性が-2mV/ のときの温度センサの出力Tempと温度の関係の一例を示している。

【 0 1 3 1 】

ここで、バンドギャップ電圧VBGRを温度によらず1200mVで一定と仮定し、このVBGRを基準電圧としてVBEをAD変換した結果が、温度を表すデジタル信号Tempとする。

50

【 0 1 3 2 】

デジタル信号Tempのビット幅は8ビットとし、1200mVを11111111（8ビット幅の最大値、10進数で255）とする。1200mVを最大値として600mVをAD変換すると、最大値の約1/2の値なので、変換結果は10000000（10進数で128）となる。基準電圧VBGRは1200mVで一定なのに対して、VBEは温度によって変化する。

【 0 1 3 3 】

例えば、300Kから+10 の310Kでは、VBEの値は $600\text{mV} - (2\text{mV}/) * 10 = 580\text{mV}$ となる。この580mVをAD変換すると、変換結果は01111100（10進数で124）となる。また、290KでのVBEは、 $600\text{mV} - (2\text{mV}/) * (-10) = 620\text{mV}$ となる。このとき温度センサ出力Tempは、例えば、10000100（10進数で132）となる。

10

【 0 1 3 4 】

図 2 1 は、ウインドウコンパレータWCMP1の回路図である。また、図 2 0 は、ウインドウコンパレータが有するフルアダーの回路図である。図 1 9 も参照しながら、図 2 0、図 2 1 の回路の動作を説明する。

【 0 1 3 5 】

図 2 0 は、図 2 1 で使用するフルアダーの回路例を示している。一般的な回路なので、動作を簡単に説明する。図 2 0 において、Cinは下位からのキャリー入力を、A、Bは加算するビット入力を、IVAn（nは整数など）はインバータを、NA3An（nは整数など）は3入力NAND回路を、NA2An（nは整数など）は2入力NAND回路を、NA4An（nは整数など）は4入力NAND回路を、Sumは加算結果を、Carryは上位へのキャリー信号を示す。

20

【 0 1 3 6 】

1ビットの全加算の真理値表は、図 2 0 中に示される。全加算器は、1ビットのA、B、Cin入力を加算して、加算結果Sumと、上位へのキャリー信号Carryを生成する。

【 0 1 3 7 】

図 2 0 に示す1ビットのフルアダーを使用して、デジタル入力の大小関係を判定するマグニチュードコンパレータを実現する。マグニチュードコンパレータを2つ組み合わせることで、あるデジタル入力THとTLとの間の範囲内にあるか、あるいは、THとTLとの間の範囲の外にあるかを判定することができる。図 2 1 の回路は、VBEデジタル値Tempの示す温度がTHとTLの示す温度間の範囲外にあるときに割り込み信号INTRRUP1を発生する。

【 0 1 3 8 】

図 2 1 のウインドウコンパレータ回路の構成と動作を簡単に説明する。図 1 9 に示したように、温度センサ出力Tempは、例えば8ビット幅の符号なしのデジタル値だとする。値の範囲は、00000000から、11111111にある。例えば、VBEは-40 から125 で、VBEが734mVから404mVに変化する。このVBEに対するAD変換結果は、10進数で156(=(734/1200)*256)から86(=(404/1200)*256)まで変化する。

30

【 0 1 3 9 】

そこで、値の大小関係を知るには、ある判定する基準の値を引いて、結果がプラスかマイナスかを判断すればよい。そのような判定は、マグニチュードコンパレータとして知られている。温度センサのデジタル変換結果Tempは、8ビット幅で符合がないが、これを、9ビット幅のデータと考えると、符号がプラスの数値として扱う。マイナスの数と加算することで、結果がプラスかマイナスかを判定すれば、Tempが所望の値より大きい小さいかを判定できる。

40

【 0 1 4 0 】

マイナスの数は、一般に2の補数として表される。引き算のための負の数を9ビット幅の2の補数として表現し、9ビット幅のプラスの数値と加算することで、引き算が実現される。一方、2の補数は、各ビット入力を反転し、最下位に1を加算すればよい。引き算は、2の補数で表現されたマイナスの数とプラスの数を加算することと等価であり、その加算結果において、最上位の9ビット目の符号ビットが0なら結果はプラス、符号ビットが1なら結果はマイナスとなる。

【 0 1 4 1 】

50

図 2 1 の左半分の回路は、8ビット幅の加算器としてはたらしき、TLの表現する温度がTempの表現する温度より高いとき、つまり、現在の温度が下限として設定した温度より低いときに、キャリビットCAR7TLをHとする。

【 0 1 4 2 】

図 2 1 において、FATL0からFATL7は、図 2 0 に示したフルアダー回路を、IVMC1からIVMC8はインバータ回路を、Temp7からTemp0は、図 1 9 で発生した温度センサ出力Tempの最上位ビットから最下位ビットのそれぞれを示している。TL7からTL0は、レジスタに設定された温度の下限TLの上位ビットから最下位ビットのそれぞれを示している。

【 0 1 4 3 】

TL7からTL0はインバータによりそれぞれビット反転され、最下位のフルアダーFATL0のキャリー入力Cinに1を加えることで、TL7からTL0は2の補数として表現される。このとき、9ビット目は、回路の実態がないが、TL7からTL0が8ビット幅の正の数なので、9ビット目は1となっている。この仮定の9ビット目が1か0かで、TempからTLを引いた結果がプラスかマイナスかを判定する。最上位の全加算器FATL7の上位へのキャリー信号CAR7TLが1になることは、仮定の9ビット目の1にさらに1を加算し、符号ビットが0になる、つまり結果がプラスとなることを意味している。

【 0 1 4 4 】

最下位のアダーFATL0のキャリー出力は、FATL1のキャリー入力に接続され、下位からのキャリーを含めて、最上位までのキャリーを判定している。それぞれのフルアダー間のキャリー信号はCARnTL (nは整数)として図示した。SUMTLn (nは整数)は各ビットの加算結果を示している。符号判定の場合は、使用しなくてもよいが、SUMTLnすべてが0のとき2つの入力一致したことが判定できる。

【 0 1 4 5 】

図 1 9 で示したように、Tempはプラスの2進数だが、温度の上昇に伴って小さくなる。従って、Tempと温度下限を示すTLの値を比べたとき、TLの値よりTempの2進数としての値が大きいとき、Tempが意味する温度は、TLより小さくなることを意味する。このため、TempからTLを引いて、その結果がプラスならCAR7TLがHとなり、これが、Tempが意味する温度がTLより小さいことをあらわしている。

【 0 1 4 6 】

同様に、2進数としてのTempは温度上昇に伴って小さくなるので、図 2 1 の右半分の回路は、THからTempを減算しているので、Tempの意味する温度がTHの意味する温度より高いときに、最上位のキャリビットCAR7THがHとなる回路となっている。

【 0 1 4 7 】

図 2 1 の右半分の回路において、FATH0からFATH7は、図 2 0 に示したフルアダー回路を、IVMC9からIVMC16はインバータ回路を、Temp7からTemp0は、図 1 9 で発生した温度センサ出力Tempの最上位ビットから最下位ビットのそれぞれを示している。TH7からTH0は、レジスタに設定された温度の上限THの上位ビットから最下位ビットのそれぞれを示している。それぞれのフルアダー間のキャリー信号はCARnTH (nは整数)として、各ビットの加算結果はSUMTHn (nは整数)として示した。

【 0 1 4 8 】

Temp7からTemp0はインバータによりそれぞれビット反転され、最下位のフルアダーFATH0のキャリー入力Cinに1を加えることで、Temp7からTemp0は2の補数として表現される。このとき、9ビット目は、回路の実態がないが、TL7からTL0が8ビット幅の正の数なので、9ビット目は1となっている。この仮定の9ビット目が1か0かで、THからTempを引いた結果がプラスかマイナスかを判定する。最上位の全加算器FATH7の上位へのキャリー信号が1になることは、仮定の9ビット目の1にさらに1を加算し、符号ビットが0になる、つまり結果がプラスとなることを意味している。

【 0 1 4 9 】

2進数としてのTHからTempを引いた結果がプラスになるということは、Tempの示す温度がTHの示す温度より高いことをあらわしている。

【 0 1 5 0 】

図 2 1 のウインドウコンパレータは、この温度が TL より低いことを示す信号 CAR7TL と、温度が TH より高いことを示す信号 CAR7TH を、OR 回路 ORMC1 で論理和とし、TOINTERRUPT1 信号を発生する。加算回路の遅延などの待ち時間を除いて値が確定した後、TOINTERRUPT1 信号が H となっているときには、温度が TL より低いのか、あるいは温度が TH より高いかを示すので、割り込み信号 INTERRUPT1 を発生し、CPU1 あるいは割り込みコントローラにより、割り込みプログラムを実行する。この点は、図 9 で説明したとおりである。

【 0 1 5 1 】

以上のように、図 2 1 に示すようなウインドウコンパレータ回路（図 6 の WCMP1）で、温度が所定温度以上変化したか否かをデジタル的に判定できる。

10

【 0 1 5 2 】

[高周波 CR 発振回路の発振周波数制御ユニット]

図 2 2 は、図 6 のカウンタ回路 COUNTER1 と TRDH 制御回路 TRDHCNTL からなる高周波 C R 発振制御ユニットの動作を示す図である。また、図 2 3 は、4MHz CR 発振回路 CROSC1 の制御信号 TRDH[7:0] と周波数の関係の一例を示す図である。図 2 3 の横軸は制御信号 TRDH[7:0] を 10 進数で表現した数を、縦軸は相対周波数を、それぞれ示している。図 6 の回路例では、CR 発振回路 CROSC1 は CROSC2 と同じ回路構成で、周波数が異なる構成の回路とした場合を示した。従って、図 2 3 の CROSC1 の周波数と制御信号の関係は、例えば、図 1 6 と同様となる。そして、周波数調整信号 TRDH[7:0] を調整して図 2 3 の縦軸の周波数比が 1 になる場合に、高周波クロック CRCK1 の周波数が 4MHz になることを意味している。

20

【 0 1 5 3 】

図 2 3 の TRDH[7:0] と周波数の関係、TRDH[7:0] がほぼ 4MHz となる初期値をフラッシュメモリ FLASH1 に記憶しておく。

【 0 1 5 4 】

図 2 2 において、FLASH1 はフラッシュメモリを、XCK2 は 32768Hz のクロックを、CRCK2 は 100kHz の CR 発振回路出力を、SEL5 はセレクタを、CKSLOW は SEL5 の出力を、ENCOUNT は CKSLOW を 1/2 に分周したクロックを、CRCK1 は 4MHz の CR クロック出力を、COUNTER1' はカウンタ回路を、PEDGNumber はカウンタでのカウント結果を、LRAM はローカルメモリを、PARAM1 は TRDH[7:0] と周波数の関係や、TRDH[7:0] の初期値など後の計算に必要なパラメータを、図 2 3 の LCPU はカウント結果と PARAM1 から TRDH[7:0] を計算する演算器あるいはシーケンサを、TRDH[7:0] は CROSC1 の制御信号をそれぞれ示している。

30

【 0 1 5 5 】

低周波側のクロック CRCK2 の H の期間 TimeH と、L の期間 TimeL は、必ずしも周期の 50% になるとは限らない。一方、クロック CRCK2 の周期は 10us 一定に制御されている。そこで、CRCK2 を例えば 1/2 に分周することで、ENCOUNT 信号は H の期間、L の期間ともに 10us となる。一方、CRCK1 の周波数が 4MHz だとすると、その周期は 250ns となる。従って、CRCK1 の周波数が正確に 4MHz だとすると、ENCOUNT 信号の周期 10us の間にクロック CRCK1 の立ち上がりは 40 回あるはずである。

【 0 1 5 6 】

そこで、たとえば 10us の ENCOUNT の H の期間を基準に、CRCK1 の立ち上がりの回数をカウンタ COUNTER1' で数えることで、CRCK1 の周波数を推測することができる。10us では 40 回立ち上がりが見られる場合と、41 回、39 回の場合の周波数の変動が大きいので、例えば、ENCOUNT の H の期間である、CRCK1 の立ち上がりを実数でカウントする期間を 100us とすれば、CRCK2 の周期から推定する CRCK1 の周期の精度を向上することができる。

40

【 0 1 5 7 】

図 2 2 では、図を単純にするために、Tcount が 10us の例を示している。所望の Tcount の間、CRCK1 の立ち上がり回数を数えて、CRCK1 の周波数を求める。ENCOUNT を L にしてカウンタ COUNTER1' をクリアして、次の計測に備えるとともに、カウント結果 PEDGNumber と、TRDH[7:0] と周波数の関係、TRDH[7:0] の現在値から、TRDH 制御回路 LCPU にて、最も望ましい TRDH[7:0] を計算する。そして、計算結果の TRDH[7:0] を CROSC1 に与えて、次の ENCOUNT

50

がHの期間で再度、周波数の誤差を計測する。これらの動作を繰り返すことで、低周波クロックCRCK2を基準にして高周波クロックCRCK1の周波数を目標の4MHzに制御することが可能となる。この高周波クロックの調整は、低周波クロックを基準として常時行われる。

【0158】

ただし、低周波CR発振回路CROSC2の調整信号TRDL[7:0]が、温度の変化が緩やかであることを期待して、割り込み処理により計算しているように、高周波CR発振回路CROSC1の周波数設定の制御も同様の割り込み処理によりCPU1で計算することも可能である。

【0159】

図22では、例えば、20us毎あるいは200us毎、110us毎といった高い頻度で、4MHzCR発振回路の制御信号TRDH[7:0]を再計算、再設定することを想定しているので、専用の計算ハードウェアLCPUとLCPUに計算のパラメータを供給するローカルメモリLRAMを使用する例を示した。TRDH[7:0]の計算に先立って、PARAM1はFLASH1からLRAMにコピーしておく。

【0160】

以上のように、図22のような回路で、常に高周波クロックCRCK1の周波数を低周波クロックCRCK2の周波数に対して一定の比となるように制御することができる。

【0161】

[低周波CR発振回路の変型例]

図24は、第1の実施の形態における低周波の100kHz発振回路の変型例を示す図である。図24の発振回路と図10の発振回路との違いは、図10のインバータIV1が、図24の発振回路では差動アンプAMP1で構成されていることにある。

【0162】

図10の発振回路の動作波形は、図1、図2と同様である。図10では、ノードND4の波形をインバータIV1が増幅してノードND1の電位が変化するノードND4の電位は、トランジスタPMA1とNMA1で決まる論理閾値となる。トランジスタPMA1とNMA1の論理閾値が電源電圧Vddの1/2に一致していれば、抵抗RV03の一端の電位はVddかGNDなので、充電、放電時に等しい電流が流れ、各部のクロックのduty比は50%になる。

【0163】

ところが、実際の回路では、トランジスタPMA1とNMA1の論理閾値は、必ずしも電源電圧Vddの1/2に一致しないので、図10の発振回路で得られるクロックのduty比は50%とはならない。

【0164】

そこで、クロックのduty比を50%に近づけておくことが望ましい場合には、図24のような発振回路とすることが望ましい。図24の発振回路では、インバータIV1として差動アンプAMP1が設けられている。そして、NBAには適切なバイアス電位を与える。さらに、抵抗RDIV1とRDIV2で電源電圧Vddを1/2に分圧し、これの分圧電位VREFを基準にして、トランジスタNMA2、NMA3によりノードND4の電位を増幅する。したがって、反転機能を有する差動アンプAMP1の閾値をVdd/2に制御することが可能となる。これにより、ノードND4がVdd/2になるタイミングで差動アンプAMP1が出力ノードND1を反転するので、出力DN1のクロックのduty比を50%に近い値にできる。

【0165】

[MCUの変型例]

図25は、第1の実施の形態におけるMCUの変型例を示す図である。図25のMCUは、クロック制御ブロックCKBLK2内の低周波CR発振回路をCROSC2'とし、さらに、抵抗のパラツキを検出するためのモニタ用抵抗RMONとを有する点で、図6のMCUと構成が異なっている。

【0166】

図25では、低周波CR発振回路をCROSC2'が、図6のCROSC2の100kHzに対して、30kHzの発振周波数を有する。つまり、リアルタイムクロック用の水晶発振器のクロックXCK2の周波数が32768Hzであることに起因して、それと近い周波数にしている。これにより、低周波CR発振回路CROSC2'のクロックを基準とする高周波CR発振回路の周波数制御と、水晶

10

20

30

40

50

発振器のクロックXCK2を基準とする周波数制御とを同等にすることができる。図25の低周波CR発振回路CROSC2'の回路構成は、図6に示したCROSC2と同等である。ただし、可変抵抗RVO2の抵抗値が異なり、発振周波数が異なる。

【0167】

次に、モニタ用抵抗RMONについて説明する。抵抗素子の温度特性は、シート抵抗の絶対値と相関がある。抵抗の温度特性は、その不純物濃度と関係があるからである。従って、使用している抵抗素子の抵抗値を知ることができれば、フラッシュメモリに格納する温度特性を抵抗値に合わせて調整することが可能となる。

【0168】

図25の回路では、この目的のために、低周波CR発振回路CROSC2'で使用している抵抗と同じ抵抗の抵抗値を測定するためのモニタ用抵抗素子RMONを有する。MCUのチップ製造後にフラッシュメモリに抵抗の温度依存性(図17の関係)を書き込む前に、外部端子PINMEASからモニタ用抵抗素子RMONの抵抗値を測定し、その測定値と関連づけて、FLASH1に発振回路の温度依存性を記憶させることが可能となる。たとえば、25でのモニタ用抵抗RMONの抵抗値に応じて、図17の関係データを調整して、若しくは複数の候補の関係データから最適なものを選択して、フラッシュメモリに書き込む。

【0169】

[第2の実施の形態]

図26は、第2の実施の形態におけるMCUの構成を示す図である。第2の実施の形態のMCUは、図6のCR発振回路に代えて、LC発振回路LCOSCを有する。そして、LC発振回路LCOSCの発振周波数を設定する周波数調整データDACIN[9:0]を、周波数制御電圧Vadjと周波数との関係データと、温度と周波数またはインダクタLの直列抵抗との関係データとに基づいて、CPUが演算で求めて、レジスタに設定する。

【0170】

図6の第1の実施の形態では、CR発振回路の温度を温度計TEMP1で測定し、CPUで発振周波数の温度変動を相殺するような周波数設定信号を求めた。図26の第2の実施の形態でも、同様の構成による周波数設定をLC発振回路LCOSCに適用する。

【0171】

図26のマイクロコントローラMCU3において、CPU1はCPUを、FLASH1はフラッシュメモリを、RAM1はSRAMを、PORT1は汎用ポートを、ADC1はAD変換回路を、MABは、CPU1、FLASH1、PORT1などをつなぐアドレスバスを、MDBは同様にCPU1、FLASH1、PORT1などをつなぐデータバスを、CKBLK3はクロック制御ブロックを、INTERRUPT2は、温度が規定の範囲を超えたときに発生される割り込み信号を、MCLKはCPUのクロックをそれぞれ示している。

【0172】

更に、図26の図6とは異なる構成として、FLCOとFLCOXはLC発振回路用の発振出力(例えば1536MHz)を、CVAとCVBは電圧制御可変容量(バラクタ)を、L1はオンチップインダクタを、PMOn(nは整数)はPMOSトランジスタを、NMOn(nは整数)はNMOSトランジスタを、Vadjはバラクタの制御電圧を、ITAIL1は発振回路のテイル電流を、DAC1はVadj電位を供給するDAC回路を、DACIN[9:0]はDACの入力信号を、REGDACINはDACIN[9:0]信号を記憶するレジスタを有する。

【0173】

また、図6と同様に、REGTH2はTHを記憶するレジスタを、REGTL2はTLを記憶するレジスタを、THは温度の上限を示す設定値を、TLは温度の下限を示す設定値を、WCMP2は温度Tempが上限THを超えるか、あるいは下限TLを下回ったときに割り込み信号INTERRUPT2を発生するウィンドウコンパレータを、TEMP1はLC発振回路の温度を知るための温度センサを、センサTEMP1内のVBEはPN接合の順方向の電圧を、TEMP1内のADCはAD変換回路を、TempはTEMP1でデジタル値に変換された温度を、REGTEMPはTempを記憶するレジスタを、ADRDACINはREGDACINのCPU1のアドレス空間内でのアドレスを、ADRTH2はREGTH2のCPU1のアドレス空間内でのアドレスを、ADRTL2はREGTL2のCPU1のアドレス空間内でのアドレスを、ADRTEMPはREGTEMPのCPU1のアドレス空間内でのアドレスを、DTDACINはCPU1とREGDACINで送受信され

10

20

30

40

50

るデータを，DTTH2はCPU1とREGTH2で送受信されるデータを，DTTL2はCPU1とREGTL2で送受信されるデータを，DTTEMPはRETEMP からCPU1へ送られるデータをそれぞれ示す。

【 0 1 7 4 】

図 2 6 において，図 6 の回路と対応する回路素子，ノード等には同じ素子名，ノード名を与えて示している。特に断らない限り，図で対応する素子，ノードには同じ名称を与えて，説明の重複を避けるものとする。

【 0 1 7 5 】

図 2 6 の LC 発振回路 LCOSC の動作を，図 2 7 ，図 2 8 ，図 2 9 ，図 3 0 ，図 3 1 を用いて説明する。図 2 7 は，バラクタの制御電圧と LC 発振回路の発振周波数の関係の一例を示す図である。図 2 8 は，バラクタの電圧が一定の場合の，LC 発振回路の周波数と温度の関係の一例を示す図である。図 2 9 は，図 2 6 の第 2 の実施の形態における MCU のメインプログラムのフローチャート図である。図 3 0 は，図 2 6 の回路の割込発生処理のフローチャート図であり，図 3 1 は，図 2 6 の回路の割り込みプログラムの処理のフローチャート図である。

【 0 1 7 6 】

図 2 7 において，横軸は制御電圧 V_{adj} を，縦軸は LC 発振回路の発振周波数を示している。例えば，バラクタを accumulation-mode のバラクタで構成しておくと，温度一定の条件下では，図 2 7 のような，制御電圧 V_{adj} が大きくなるのと同時に発振周波数が大きくなる特性が得られることが知られている。この理由は，制御電圧 V_{adj} の増大に応じてバラクタの容量が減少し，LC 共振回路の共振周波数が増大するためである。

【 0 1 7 7 】

LC 発振回路は，製造時のばらつきにより容量の絶対値が大きい場合，小さい場合，中間の場合でも，発振周波数は変化する。図 2 7 の 3 本の線は容量 C が（上から）小さい場合，中心の場合，大きい場合の特性を示している。したがって，製造後に発振周波数をチェックすることによりどの特性がマッチングしているかを決定し，その特性をフラッシュメモリに記憶するのが望ましい。

【 0 1 7 8 】

一方，図 2 8 は，制御電圧 V_{adj} を一定としたときの，相対周波数（0%が中心）と温度の関係を示している。ここで，相対周波数は，目標とする周波数 1536MHz との差分の比率である。図示されるとおり，バラクタの制御電圧 V_{adj} を一定としたとき，LC 発振回路の発振周波数は温度上昇にともなって，（数%ではあるが）小さくなることが知られている。配線で作られたオンチップインダクタの直列抵抗が温度上昇とともに大きくなることから，周波数が温度上昇とともに小さくなる原因である。また，容量のキャパシタンス（容量値）とインダクタ $L1$ のインダクタンスは，温度変動によって変動することはない。温度変動により変動するのは，インダクタの直列抵抗である。

【 0 1 7 9 】

図 2 6 の MCU では，図 6 の回路とほぼ同様の構成で，LC 発振回路 LCOSC の発振周波数の温度変動を補償する。つまり，電源起動時などの初期化時は，フラッシュメモリに記憶した発振周波数制御信号に対応する制御電圧と発振周波数との関係データ（図 2 7 ）と，温度に対する発振周波数（またはインダクタの直列抵抗）との関係データ（図 2 8 ）と，現在の温度との関係から，CPU が，目標とする発振周波数に調整できる周波数調整信号に対応する制御電圧 V_{adj} を演算で求める。また，温度が変動した場合には，ウインドウコンパレータが割込信号を発生し，それに応答して，CPU が初期化時と同様にして制御電圧 V_{adj} を演算で求める。

【 0 1 8 0 】

LC 発振回路は，単独で 1 ~ 15GHz の周波数帯域で発振できる。したがって，図 2 6 では，図 6 のように 2 つの CR 発振回路を組み合わせるのではなく，単独の LC 発振回路を設け，その発振周波数を目標値にするための制御電圧 V_{adj} を CPU による演算で求める。

【 0 1 8 1 】

[LC 発振器の周波数調整のメインプログラム]

10

20

30

40

50

図29のメインプログラムは、MCUに電源が供給されて、リセットが解除されると実行される。メインプログラムの実行開始（スタート）後、図26内の温度センサTEMP1が、インダクタL1の温度を測定する（STEPM1）。温度センサTEMP1は図6と同様の構成であり、温度に依存したVBEをADCして温度に対応した温度 - デジタル変換結果Tempが出力される。ここでは、STEPM1で、仮にTemp=T1度だったものとする。

【0182】

次に、STEPM2では、フラッシュメモリFLASH1に記憶した図27の関係データから、例えば温度25のときに出力周波数が目標周波数となる周波数調整信号DACIN[9:0]を読み出す。そして、STEPM3では、フラッシュメモリFLASH1から、図28の温度と発振周波数の関係データを読み出す。

10

【0183】

図29のSTEPM1、STEPM2、STEPM3で、現在のL1の温度、温度が25のときの周波数制御信号と周波数の関係（図27）、周波数制御信号DACIN[9:0]を固定したときの周波数と温度の関係（図28）が得られる。そこで、STEPM4では、図28から、現在温度T1のときの周波数の低下の割合が分かるので、図27の関係から、温度により下がる周波数を相殺するための制御電位Vadj（またはそれに対応する周波数調整信号DACIN[9:0]）を演算により求める。つまり、図27と図28からLC発振回路の出力を（例えば）1536MHzに保つための制御電位Vadjの値を計算で求める。この計算はCPUがプログラムを実行することで行われる。

【0184】

20

次に、STEPM5では、上でもとめた周波数調整信号DACIN[9:0]をレジスタREGDACINに書き込む。また、現在温度T1に対して、例えば10以上温度が低くなったとき、あるいは、10以上温度が高くなったとき、再度周波数調整信号DACIN[9:0]の値を再計算するために、割り込み信号を発生できるように、T1-10をレジスタREGTL2に、T1+10をレジスタREGTH2にそれぞれ書き込む。例えば、T1が70、再設定しない温度の範囲を+-10とすれば、THは80、TLは60となる。

【0185】

上記の図29のメインプログラムの初期化を実行した後に、MCUユーザーのユーザープログラムの実行を開始する（STEPM6）。

【0186】

30

[割込信号の発生]

図30は、割込発生動作のフローチャート図である。LC発振回路を使用する場合には、温度センサTEMP1を動作させて、インダクタL1の温度を監視する。図29の初期化で説明した初期化時の温度T1が70、上限温度THは80、下限温度TLは60の例で説明すると、温度センサTEMP1の温度計測結果Tempが60から、80の範囲内にあるときは、STEPM1とSTEPM2を繰り返す。STEPM1で温度を測り、STEPM2で温度が60から、80の範囲内にあるかを判定する。温度がTL=60より低いか、TH=80より高い場合、STEPM3に実行が移り、ウィンドウコンパレータがCPUへの割り込み信号を発生する。そして、割り込みコントローラが、この周波数調整割込信号INTERRUPT2を受信すると、CPUは、メインプログラムから割込プログラムに実行を移す。

40

【0187】

[割り込み時の周波数調整処理]

図31は、割り込みプログラムでの処理のフローチャート図である。図31のSTEPM11で、温度センサTEMP1が現在温度に対するデジタル値Tempを出力する。ここでは仮にTemp=TX度だったものとしている。割り込み信号INTERRUPT2が発生されているので、温度TempはTL=60より低いか、TH=80より高い温度となっている。ここでは仮にTemp=TX=90とする。CPUは、アドレスADRTEMPのレジスタREGTEMPを読み出すことで、測定された現在温度に対応するデジタル値Tempを読み出す。

【0188】

次に、CPUは、STEPM12で、フラッシュメモリFLASH1から、温度25のときの周波数調整

50

信号DACIN[9:0]と発振周波数の関係を示す図27のデータを読み出す。さらに、CPUは、STEMPI3ではフラッシュメモリFLASH1から温度と発振周波数の関係を示す図28の関係データを読み出す。そして、STEP14で、出力周波数を目標とする周波数に保つための周波数調整信号DACIN[9:0]の値を計算で求める。この演算は、初期化处理と同様であり、図28の関係データから現在の温度に対する周波数の変動を取得し、それに基づいて、図27のデータから目標の周波数にするための周波数調整信号DACIN[9:0]を計算する。

【0189】

そして、STEP15では、上でもとめたDACIN[9:0]をレジスタREGDACINに書き込む。また、現在温度TX=90 に対して、例えば10 以上温度が低くなったとき、あるいは、現在温度TX=90 に対して10 以上温度が高くなったとき、割り込み信号を発生して、再度DACIN[9:0]の値を再計算することができるように、TX-10 をレジスタREGTL2に、TX+10 をレジスタREGTH2にそれぞれ書き込む。例えば、TXが90 ，再設定しない温度の範囲を+-10 とすれば、THは100 ，TLは80 となる。

10

【0190】

以上のように、周波数調整信号DACIN[9:0]の再計算と、割り込み信号発生の変更条件の変更を実行して、プログラムの実行をメインプログラムにもどす。周波数調整信号DACIN[9:0]は、DAC1でアナログ値である制御電圧Vadjに変換され、その制御電圧Vadjがバラクタの電極に印加される。

【0191】

以上説明したように、図29、図30、図31の処理と、図26のハードウェアを使用して、インダクタL1の温度を監視し、その温度がある範囲を超えたときに、周波数調整信号を再度計算し、発振回路出力が目標周波数にもっとも近くなるように制御することが可能となる。

20

【0192】

[第3の実施の形態]

図32は、第3の実施の形態におけるMCUの構成図である。このMCUは、図6のMCUの低周波CR発振回路CROSC2と、図26のMCUのLC発振回路LCOSCとを有し、それらの発振回路は、図6、図26と同様に、温度の変動に応じて最適な周波数への調整が、CPUによるプログラムの実行によって行われる。

【0193】

簡単のために、図26のMCUと異なる部分のみ説明する。MCU4内において、CKBLK4はクロック制御ブロックを示している。LC発振回路のバラクタの制御電圧Vadjは周波数調整信号DACIN[9:0]をDACデジタルアナログ変換して発生する。この周波数調整信号DACIN[9:0]を、初期化時に温度を測定してCPUによる計算プログラムの実行で求め、温度が変化したことを検出して割り込みINTERRUPT2を発生し、再度CPUによる計算プログラムの実行で周波数調整信号DACIN[9:0]を再計算することは、図26の回路と同じである。図32の回路ではさらに、図6の100kHzCR発振回路CROSC2も集積している。CROSC2の周波数制御に周波数調整信号TRDL[7:0]を使用することは、図6と同じである。初期化時に温度を測定して周波数調整信号TRDL[7:0]を求め、温度が変化したことを検出して割り込みINTERRUPT1を発生し、周波数調整信号TRDL[7:0]を再計算することも図6と同じである。

30

40

【0194】

また、CR発振回路CROSC2とLC発振回路LCOSCの再設定のトリガに使用する温度計TEMP1は、L1の温度と、CROSC2の温度に近い場合には共通の温度センサTEMP1で検出することができる。

【0195】

図32の第3の実施の形態では、クロック制御ブロックCKBLK4内に、高周波LC発振回路LCOSCと、低周波CR発振回路CROSC2とを有し、それらが生成する高周波クロックと低周波クロックとを、図6に示したセレクタ、PLL、分周器などの組み合わせにより生成したクロックを、CPUの同期クロックとして使用する。しかも、両クロックの周波数は、温度変化にかかわらず、常に目標とする周波数に制御される。

50

【 0 1 9 6 】

以上、幾つかの実施の形態で説明したとおり、CR発振回路またはLC発振回路の発振周波数を、CPUが、周波数調整プログラムを実行することで、発振周波数対周波数調整信号の関係データと、温度対発振周波数の関係データと、測定した現在温度とに基づいて、目標の発振周波数を得るための周波数調整信号を求めて、発振回路に設定する。したがって、複雑なハードウェア回路を用いることなく、発振回路の周波数の温度依存性を抑制することができる。

【 0 1 9 7 】

以上の実施の形態をまとめると、次の付記のとおりである。

【 0 1 9 8 】

(付記1)

CPUと、

周波数調整信号に応じて周波数が可変制御されるクロックを生成して前記CPUに供給する発振回路と、

温度を検知する温度センサと、

前記温度センサにより検知される温度が所定温度変動したことに応答して前記CPUにより実行される周波数調整プログラムと、前記周波数調整信号と前記発振回路の発振周波数との関係を示す調整信号対周波数関係データと、温度と前記発振回路の発振周波数との関係を示す温度対周波数関係データとを格納するメモリとを有し、

前記周波数調整プログラムが前記CPUにより実行されることで、前記CPUが、前記温度対周波数関係データと調整信号対周波数関係データとに基づいて、前記温度センサにより検知される現在温度に応じて、前記発振回路の発振周波数を目標の周波数に制御する周波数調整信号を演算し、

前記演算した周波数調整信号が前記発振回路に設定されるマイクロコントローラ。

【 0 1 9 9 】

(付記2)

付記1において、

更に、前記現在温度から前記温度センサにより検知される温度が所定温度変化したことを検知して前記CPUに周波数調整割込信号を出力する温度変化検出ユニットを有し、

前記CPUは前記周波数調整割込信号に応じて前記周波数調整プログラムを実行するマイクロコントローラ。

【 0 2 0 0 】

(付記3)

付記1または2において、

前記CPUは、電源起動時を含むリセット時にも、前記周波数調整プログラムを実行して前記周波数調整信号を演算し、当該演算した周波数調整信号が前記発生回路に設定されるマイクロコントローラ。

【 0 2 0 1 】

(付記4)

付記1または2において、

前記発振回路は、リング状に接続された複数のインバータ回路と、前記リング状の接続経路内に設けられた抵抗及びキャパシタとを有し、前記抵抗の抵抗値が前記周波数調整信号に応じて可変設定されるCR発振回路であり、

前記CR発振回路は、第1の周波数調整信号に応じて第1の発振周波数で発振する低周波CR発振回路と、第2の周波数調整信号に応じて前記第1の発振周波数より高い第2の発振周波数で発振する高周波CR発振回路とを有し、

前記第1の周波数調整信号は、前記周波数調整プログラムを前記CPUが実行することで演算され、前記低周波CR発振回路に設定され、

更に、前記第1の周波数調整信号により調整された前記低周波CR発振回路の第1の発振周波数と、前記高周波CR発振回路の第2の発振周波数との周波数比が、前記目標の周

10

20

30

40

50

波数に対応する周波数比になるように，前記第2の周波数調整信号を生成し，前記第2のCR発振回路に設定する高周波CR発振制御ユニットを有し，

前記第2の高周波CR発振回路の出力クロックが前記CPUに供給されるマイクロコントローラ。

【0202】

(付記5)

付記4において，

更に，前記第2の発振周波数より低い第3の発振周波数で発振する第1の水晶発振回路用の第1の水晶振動子を外部に接続可能な第1の水晶振動子接続端子を有し，

前記高周波CR発振制御ユニットは，前記低周波CR発振回路の第1の発振周波数のクロックに代えて，前記第3の発振回路の第3の発振周波数と，前記第2の発振周波数との周波数比が，前記目標の周波数に対応する周波数比になるように，前記第2の周波数調整信号を生成するマイクロコントローラ。

10

【0203】

(付記6)

付記4または5において，

更に，前記第2の発振周波数で発振する第2の水晶発振回路用の第2の水晶振動子を外部に接続可能な第2の水晶振動子接続端子を有し，

前記高周波CR発振回路の出力クロックと，前記第2の水晶発振回路の出力クロックとのいずれかを選択する選択回路を有するマイクロコントローラ。

20

【0204】

(付記7)

付記4または5において，

前記温度対周波数関係データは，温度と前記CR発振回路の抵抗の抵抗値との関係を有し，

更に，外部端子に接続され，外部から基準温度における抵抗値が測定可能なモニタ抵抗を有し，

前記温度対周波数関係データは，前記測定された基準温度における抵抗値に応じて設定されるマイクロコントローラ。

【0205】

(付記8)

付記1または2において，

前記発振回路は，互いに接続されたインダクタとキャパシタを有し，前記キャパシタの容量値が前記周波数調整信号に応じて可変設定されるLC発振回路であるマイクロコントローラ。

30

【0206】

(付記9)

付記1または2において，

前記発振回路は，リング状に接続された複数のインバータ回路と，前記リング状の接続経路内に設けられた抵抗及びキャパシタとを有し，前記抵抗の抵抗値が前記周波数調整信号に応じて可変設定されるCR発振回路であるマイクロコントローラ。

40

【符号の説明】

【0207】

CROSC1：高周波CR発振回路

CROSC2：低周波CR発振回路

TRDH[7:0]：高周波発振周波数調整信号

TRDL[7:0]：低周波発振周波数調整信号

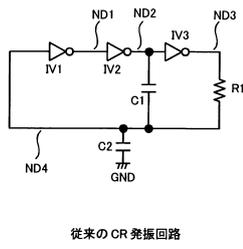
TEMP1：温度センサ

WCMP1：ウィンドウコンパレータ

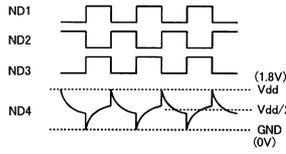
INTERRUPT1：周波数調整割込信号

COUNTER1, TRDHCONTL：高周波CR発振制御ユニット

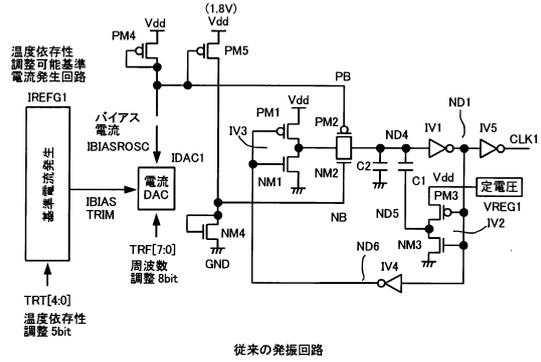
【図1】



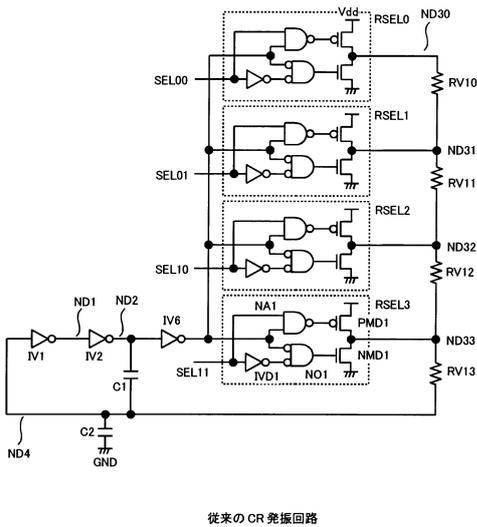
【図2】



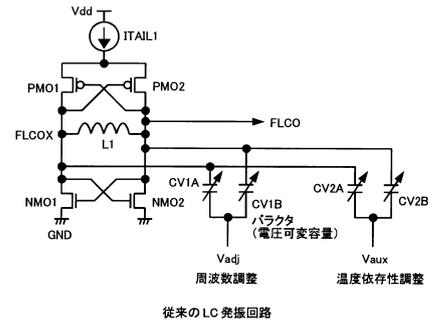
【図3】



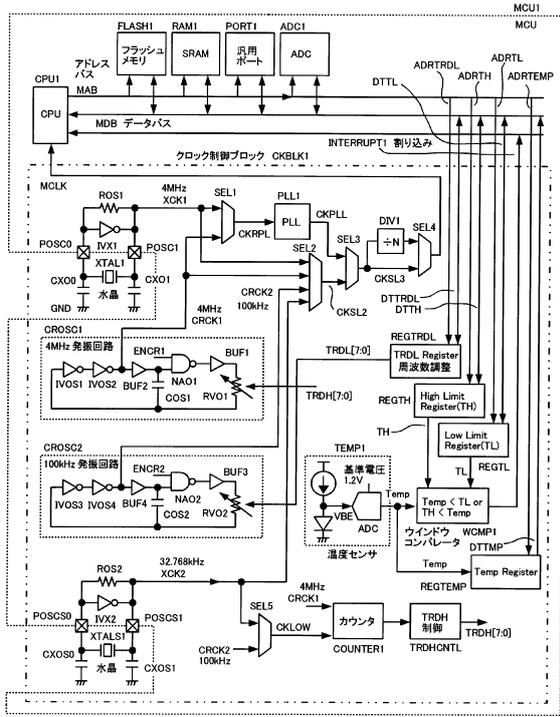
【図4】



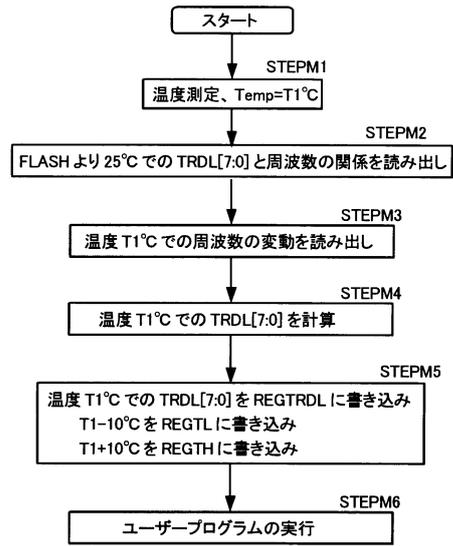
【図5】



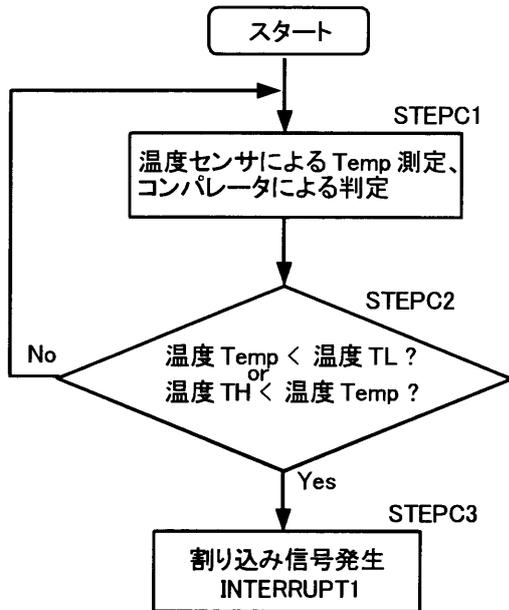
【図6】



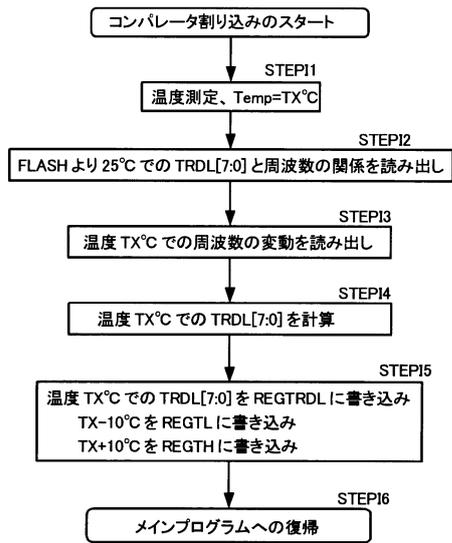
【図7】



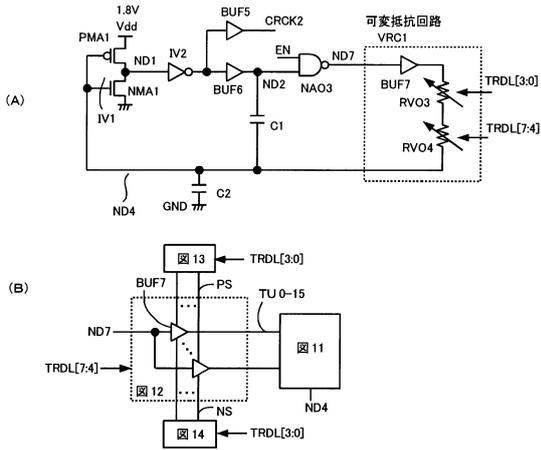
【図8】



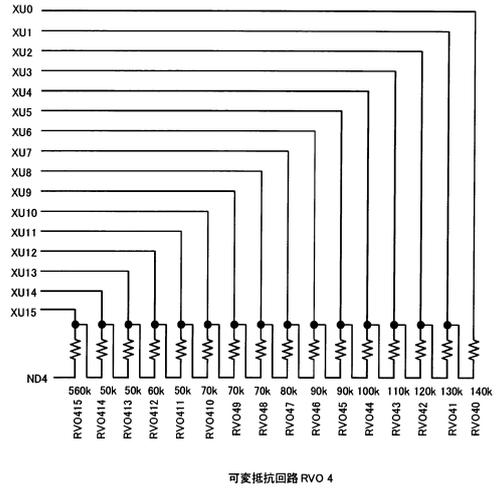
【図9】



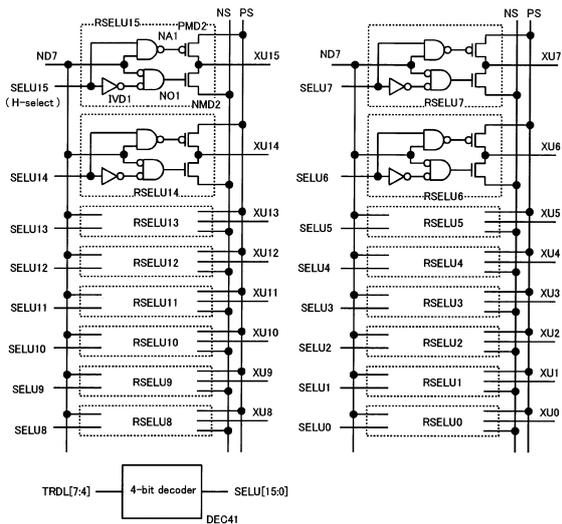
【 図 1 0 】



【 図 1 1 】

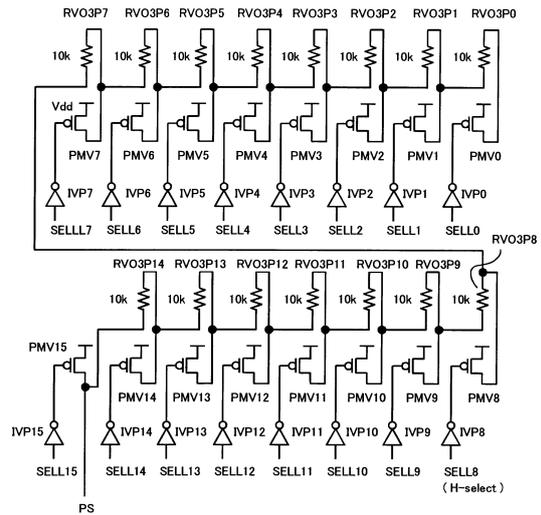


【 図 1 2 】



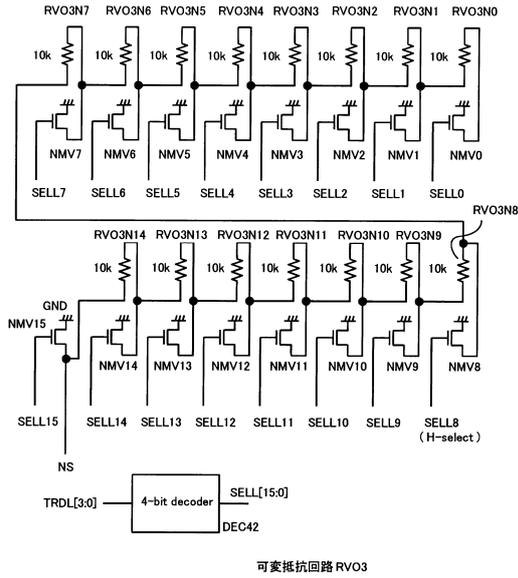
可変抵抗回路 RVO4, BUF7

【 図 1 3 】

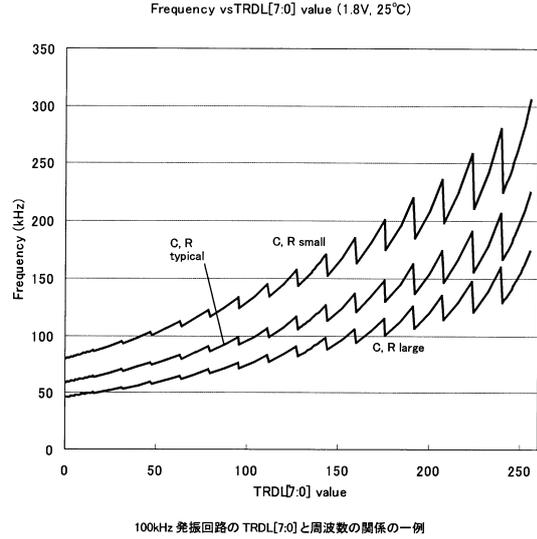


可変抵抗回路 RVO3

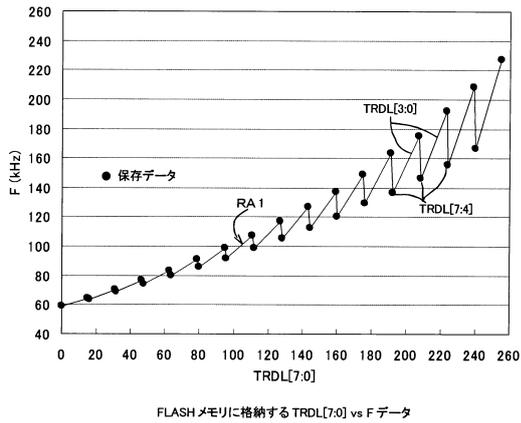
【 図 1 4 】



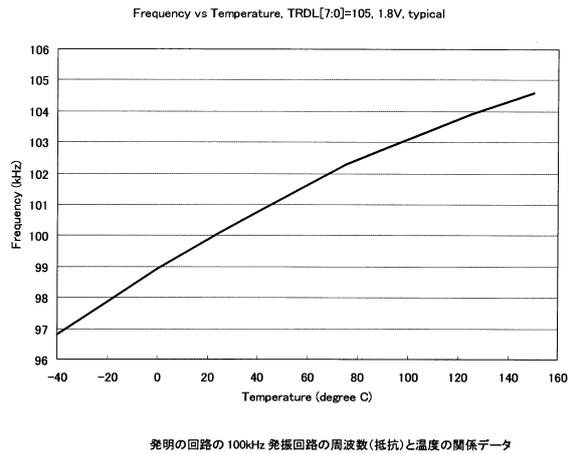
【 図 1 5 】



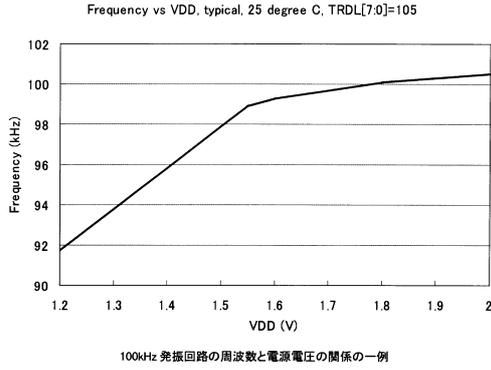
【 図 1 6 】



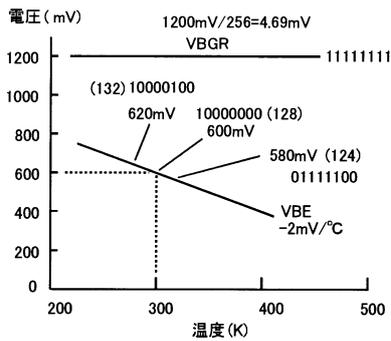
【 図 1 7 】



【図18】

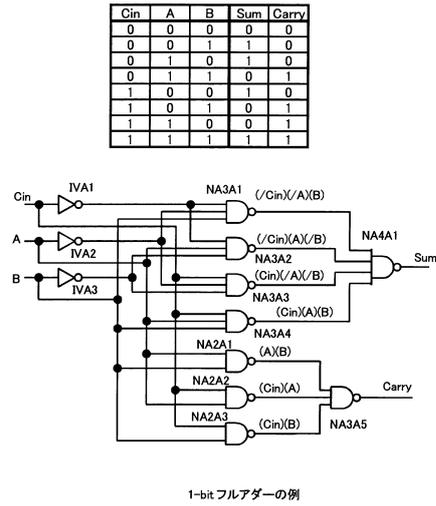


【図19】

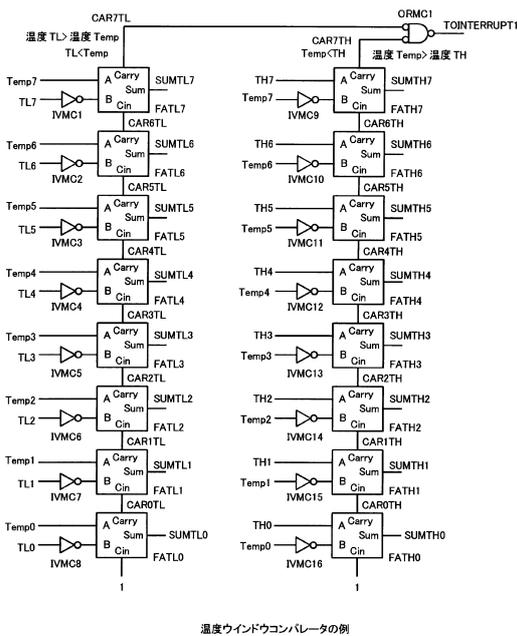


温度センサの例 (VBE の値と AD 変換結果の一例)

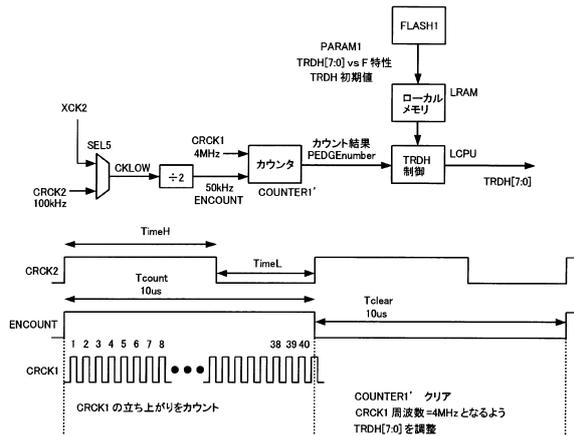
【図20】



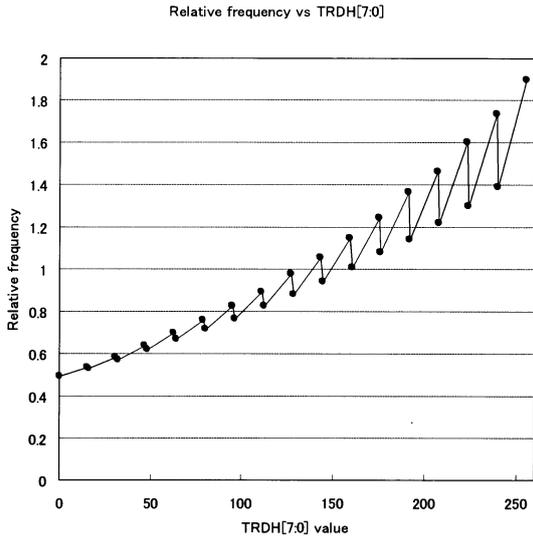
【図21】



【図22】

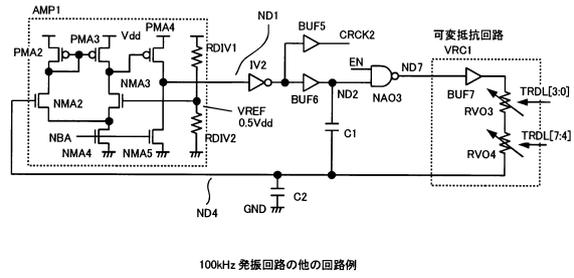


【図 2 3】



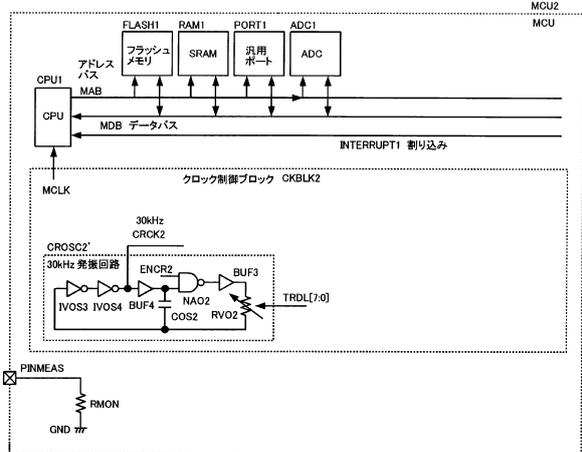
4MHz 発振回路の周波数と TRDH[7:0] の関係の一例

【図 2 4】

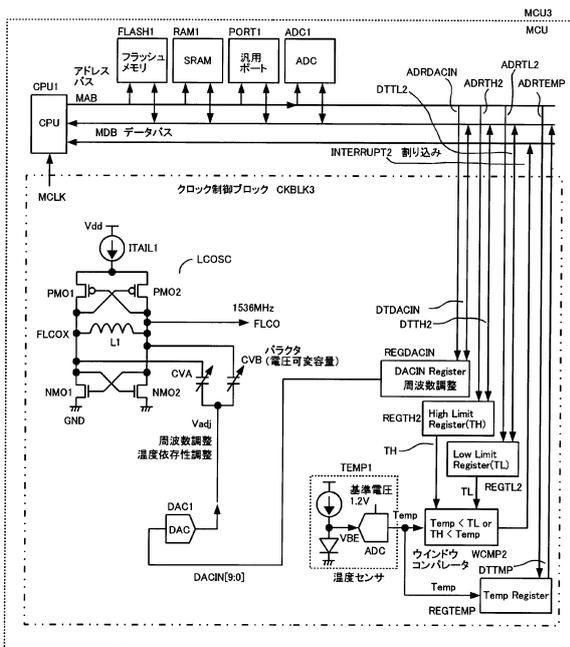


100kHz 発振回路の他の回路例

【図 2 5】

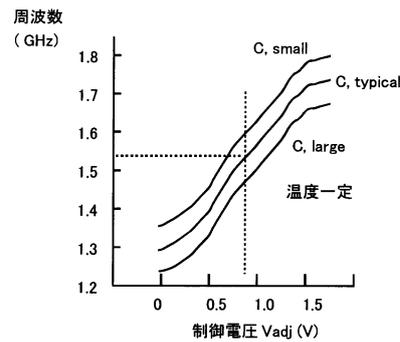


【図 2 6】



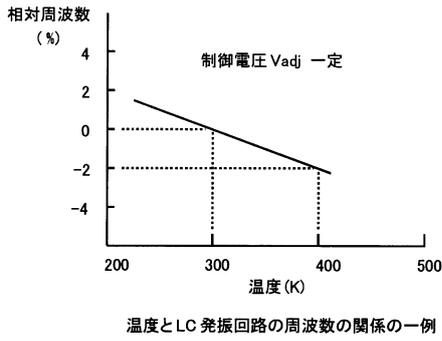
第2の実施の形態

【図 2 7】

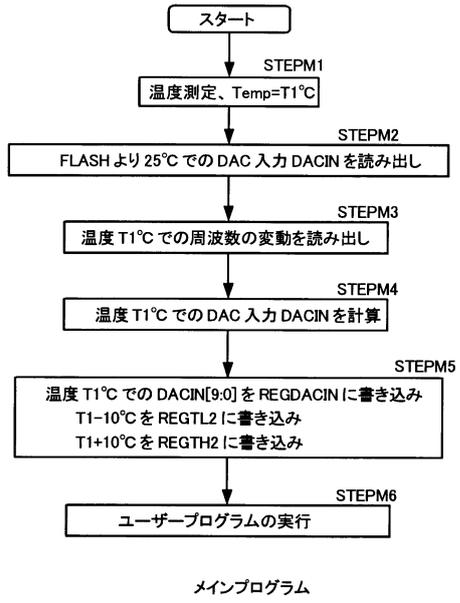


バラクタの制御電圧と周波数の関係の一例

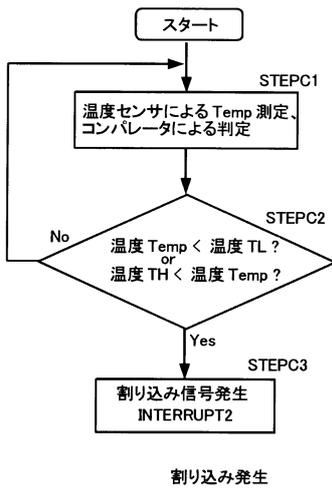
【図28】



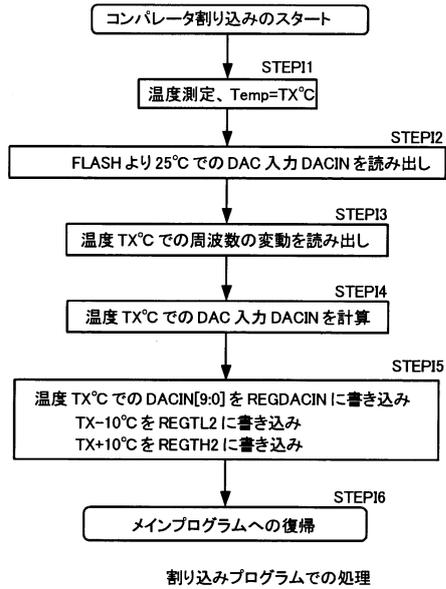
【図29】



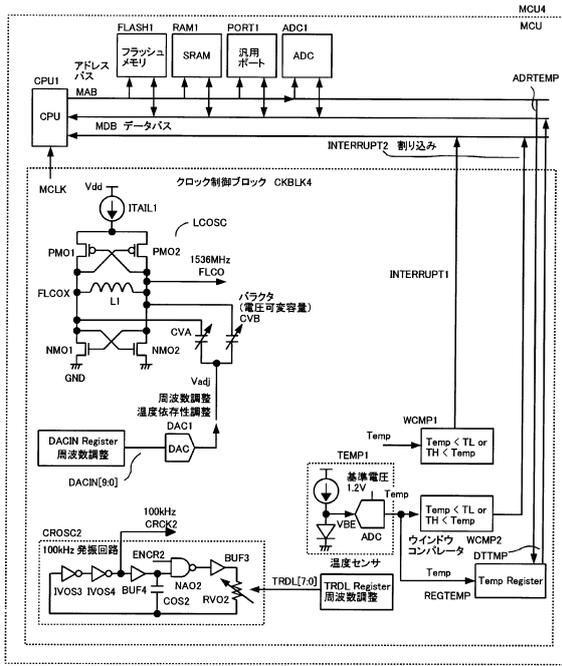
【図30】



【図31】



【図 3 2】



第3の実施の形態

フロントページの続き

- (72)発明者 谷田 行庸
神奈川県横浜市港北区新横浜二丁目10番23 富士通セミコンダクター株式会社内
- (72)発明者 石川 清
神奈川県横浜市港北区新横浜二丁目10番23 富士通セミコンダクター株式会社内
- (72)発明者 伊藤 雄作
神奈川県横浜市港北区新横浜二丁目10番23 富士通セミコンダクター株式会社内
- (72)発明者 大倉 鉄郎
神奈川県横浜市港北区新横浜二丁目10番23 富士通セミコンダクター株式会社内
- (72)発明者 田中 正博
神奈川県横浜市港北区新横浜二丁目10番23 富士通セミコンダクター株式会社内

審査官 白井 孝治

- (56)参考文献 特開2002-204127(JP,A)
特開2000-124736(JP,A)
特開2007-194711(JP,A)
特開平10-333789(JP,A)

(58)調査した分野(Int.Cl., DB名)

H03K 3/00~ 3/356
H03B 5/00~ 5/42
G06F 1/04~ 1/14