

基本特性

- 分辨率: 14 位
- 2.5GSPS 时钟频率, 支持直接射频输出
 - 普通模式 DC 到 1.25GHz
 - 混频模式 1.25GHZ 到 3GHz
- SFDR \geq 55dBc@fout=950MHz
- 双端口 LVDS 数据接口
 - 每端口支持高达 1.25GSPS 数据输入
 - 源同步 DDR 驱动
- 支持多芯片同步功能
- 8.7mA 到 31.7mA 的可编程输出电流

应用领域

- 宽带通信系统
- 军用宽带电子系统
- 仪器、自动测试设备
- 雷达、航空设备
- CMTS 系统设备

产品描述

CBM97D39BG 是 14bit 2.5GSPS 的高性能射频 DAC, 可以直接产生 DC 到 3GHz 的信号。其 DAC 内核采用四相开关结构, 从而能提供优越的低失真性能以及业界领先的直接射频输出能力。芯片工作在基带模式时, 能在第一奈奎斯特频率内产生的多载波宽带信号; 工作在混频模式时, 能在第二、三奈奎斯特区域内输出多载波信号。另外, DAC 的输出电流范围从 8.7mA 到 31.7mA 可调整。片上控制器大大简化了系统设计, 芯片具有双端、源同步的 LVDS 接口, 从而简化了与 FPGA/ASIC 之间的数字接口。片上控制器能在大温度变化范围内管理内部和外部时钟域之间的接口, 从而保证数据从主机到 DAC 内核的正确传输。多芯片同步功能通过片上同步控制器实现。芯片通过串行外围接口 (SPI) 来进行配置和寄存器访问。

CBM97D39BG 采用 0.18um CMOS 工艺实现, 工作在 1.8V 和 3.3V 双电源。封装采用 160 脚 CSBGA。

功能框图

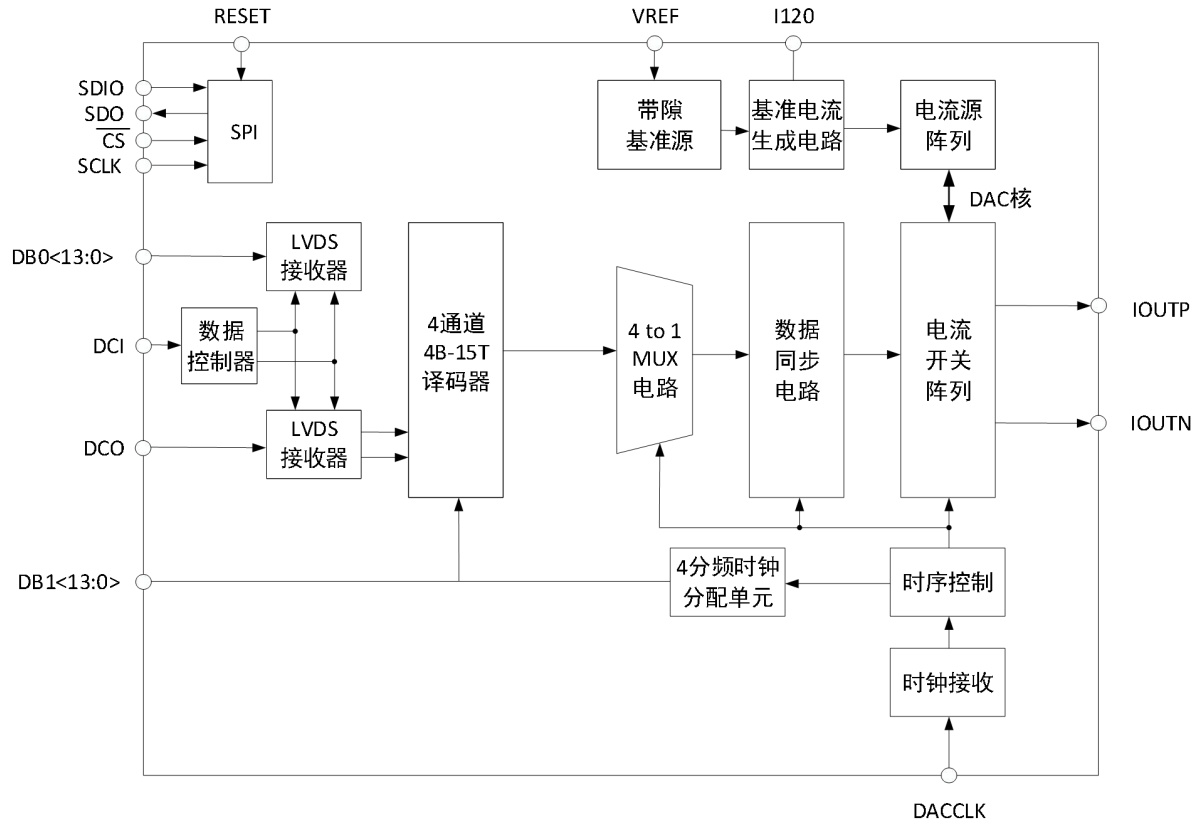


图 1. CBM97D39 功能框图

目 录

基本特性	1
应用领域	1
产品描述	1
功能框图	1
目录	3
修订日志	3
产品特性	6
性能指标	6
最大额定值	7
热力学阻抗	7
管脚说明	8
典型性能特征	12
普通模式输出	12
混频模式输出	18
串行控制 (SPI) 寄存器	20
复位	20
SPI 操作	20
SPI 指令格式	20
SPI 寄存器	23
SPI 端口配置和软件复位寄存器	26
LVDS 接口和 DAC 断电控制寄存器	26
控制时钟使能寄存器	26
中断请求 (IRQ) 使能/状态	26
DAC 满量程电流设置 (I_{OUTFS}) 和睡眠控制寄存器	27
DAC 四项-开关工作模式寄存器	28
DCI 相位调整状态	28
SYNC_IN 相位调整状态	28
数据接收控制寄存器配置	28
数据接收控制器_数据采样延时寄存器	29
数据 SYNC 接收机控制器_DCI 延时值/窗口和相位切换寄存器	29
数据接收机控制器_延迟线状态和 SYNC 控制器 SYNC_ 状态寄存器	30
SYNC 和数据接收机控制器锁定/跟踪状态寄存器	31
CLK 接收机控制寄存器	31
MU 控制器配置和状态寄存器	32
器件标识寄存器	33
工作说明	33

LVDS 数据接收	33
数据接收机初始化	36
数据接收机工作在低频时钟的情况下	37
LVDS 驱动和接收电路	38
MU 控制器	39
MU 控制器初始化描述	43
中断请求	44
多芯片同步	46
同步控制器的初始化说明	49
同步限制	49
模拟接口	50
模拟工作模式	50
时钟输入	53
电压基准	54
模拟输出	55
DAC 的等效输出电路和传递函数	55
DAC 的峰值输出功率	56
输出级配置	57
启动过程推荐	59
封装尺寸	63
包装/订购信息	64

修订日志

版本	修订日期	变更内容	变更原因	制作	审核	备注
V1.0	2025.2.24	更新产品特性参数，功能框图统一	错误更新	WW	LYL	

产品特性

- 1、可在第一、第二奈奎斯特区间内，输出带宽高达 1.25GHz 的高质量宽带信号；
- 2、四相开关 DAC 内核结构提供了优越的 AC 线性度，同时支持 DAC 工作在混频模式；
- 3、双端、DDR、LVDS 接口，支持最大 2.5GSPS 的转换器；
- 4、通过片上控制器管理内外时钟域之间的相差；
- 5、多芯片同步功能；
- 6、可编程差分输出电流，从 8.7mA 到 31.7mA。

性能指标

除非另有说明，工作条件：VDDA = VDD33 = 3.3 V，VDDC = VDD = 1.8 V，温度：-40~85°C， $f_{CLK}=2500\text{MHz}$ 。

表 1

参数	符号	条件	最小值	典型值	最大值	单位
分辨率	RES			14		Bits
基准电压	V_{REF}		1.15	1.18	1.25	V
线性误差	E_L		-3	± 2.5	3	LSB
微分线性误差	E_{DL}		-3	± 0.8	3	LSB
增益误差	E_G		-8	3	8	%FSR
功耗	P_D		--	0.95	1.25	W
无杂散动态范围	SFDR	fo=100MHz, normal mode	62	70	--	dBc
		fo=350MHz, normal mode	52	59	--	dBc
		fo=550MHz, normal mode	51	56	--	dBc
		fo=950MHz, normal mode	46	57	--	dBc
		fo=1400MHz, normal mode	44	47	--	dBc
		fo=1900MHz, normal mode	50	53	--	dBc
		fo=2400MHz, normal mode	44	45	--	dBc

最大额定值

参数	相比较于	范围
VDDA	VSSA	-0.3V到+3.6V
VDD33	VSS	-0.3V到+3.6V
VDD	VSS	-0.3V到+1.98V
VDDC	VSSC	-0.3V到+1.98V
VSSA	VSS	-0.3V到+0.3V
VSSA	VSSC	-0.3V到+0.3V
VSS	VSSC	-0.3V到+0.3V
DACCLK_P,DACCLK_N	VSSC	-0.3V到VDDC+0.18V
DCI,DCO,SYNC_IN,SYNC_OUT	VSS	-0.3V到VDD33+0.3V
LVDS数据输入	VSS	-0.3V到VDD33+0.3V
IOUP,IOUTN	VSSA	-1.0V到VDDA+0.3V
I120,VREF	VSSA	-0.3V到VDDA+0.3V
IRQ,SCLK,SDO,SDIO,RESET	VSS	-0.3V到VDD33+0.3V
结温		150°C
存储温度		-65°C到+150°C

热力学阻抗

θ_{JA} 特指在最差条件下，即器件焊接在表面封装的电路板上。

封装类型	θ_{JA}	θ_{JC}	单位
160-Ball BGA	31.2	7.0	°C/W ⁵

⁵ 没有空气流动

管脚说明

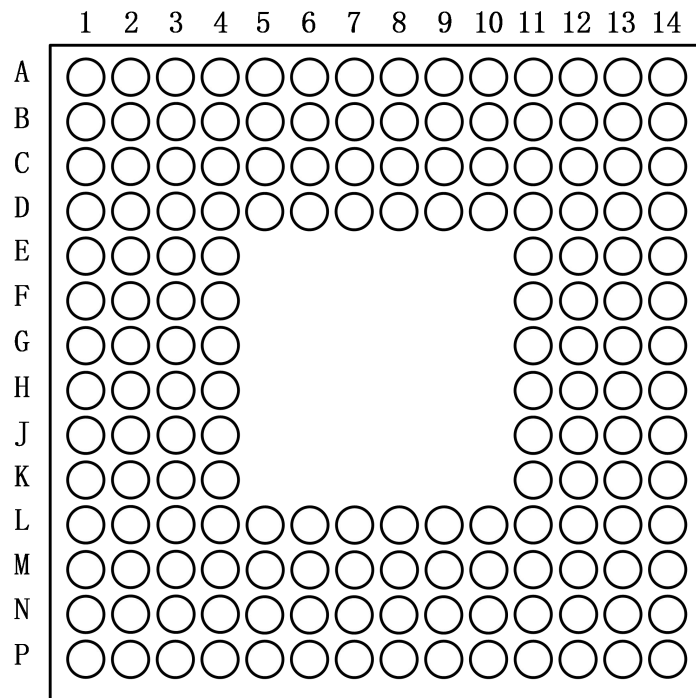


图 2 CBM97D39BG 管脚图 (俯视图, 从器件顶部向下看的对应关系)

端口号	端口符号	特性说明				
		模式	I/O	电压范围	最大频率	其他
C1,C2,D1,D2,E1,E2,E3,E4	VDDC	/	/	1.8V	/	/
A1,A2,A3,A4,A5,B1,B2,B3,B4,B5,C4,C5,D4,D5	VSSC	/	/	0V	/	/
A10,A11,B10,B11,C10,C11,D10,D11	VDDA	/	/	3.3V	/	/
A12,A13,B12,B13,C12,C13,D12,D13	VSSA	/	/	0V	/	/
A6,A9,B6,B9,C6,C9,D6,D9,F1,F2,F3,F4,E11,E12,E13,E14,F11,F12	VSSA Shield	/	/	0V	/	/
A14	NC	/	/	/	/	/
A7,B7,C7,D7	IOUTN	模拟	O	3.0V	3.6GHz	/
A8,B8,C8,D8	IOUTP	模拟	O	3.3V, 10~30m	3.6GHz	/

				A满幅输出 电流		
B14	I120	模拟	I/O	约1.2V	/	/
C14	VREF	模拟	I/O	约1.2V	/	/
D14	VREF	模拟	O	3.3V, 约 10 μ A电流 输出	/	/
C3,D3	DACCLK_N/DAC CLK_P	模拟	I	1.8V	2.5GHz	/
F13	IRQ	数字	O	3.3V	/	/
F14	RESET	数字	I	3.3V	/	/
G13	/CS	数字	I	3.3V	/	/
G14	SDIO	数字	I/O	3.3V	/	/
H13	SCLK	数字	I	3.3V	/	/
H14	SDO	数字	O	3.3V	/	/
J3,J4,J11,J12	VDD33	/	/	3.3V	/	/
G1,G2,G3,G4,G11,G12	VDD	/	/	1.8V	/	/
H1,H2,H3,H4,H11,H12,K3,K4, K11,K12	VSS	/	/	0V	/	/
J1,J2	SYNC_OUT_P/S YNC_OUT_N	数字	O	1.8V	625MHz	/
K1,K2	SYNC_IN_P/SYN C_IN_N	数字	I	1.8V	625MHz	/
J13,J14	DCO_P/DCO_N	数字	O	1.025V~1 .375V	625MHz	/
K13,K14	DCI_P/DCI_N	数字	I	0.825V~1 .575V	625MHz	/
L1,M1	DB1[0]P/DB1[0] N	数字	I	0.825V~1 .575V	1.25GSPS	/
L2,M2	DB1[1]P/DB1[1] N	数字	I	0.825V~1 .575V	1.25GSPS	/

L3,M3	DB1[2]P/DB1[2] N	数字	I	0.825V~1 .575V	1.25GSPS /
L4,M4	DB1[3]P/DB1[3] N	数字	I	0.825V~1 .575V	1.25GSPS /
L5,M5	DB1[4]P/DB1[4] N	数字	I	0.825V~1 .575V	1.25GSPS /
L6,M6	DB1[5]P/DB1[5] N	数字	I	0.825V~1 .575V	1.25GSPS /
L7,M7	DB1[6]P/DB1[6] N	数字	I	0.825V~1 .575V	1.25GSPS /
L8,M8	DB1[7]P/DB1[7] N	数字	I	0.825V~1 .575V	1.25GSPS /
L9,M9	DB1[8]P/DB1[8] N	数字	I	0.825V~1 .575V	1.25GSPS /
L10,M10	DB1[9]P/DB1[9] N	数字	I	0.825V~1 .575V	1.25GSPS /
L11,M11	DB1[10]P/DB1[1 0]N	数字	I	0.825V~1 .575V	1.25GSPS /
L12,M12	DB1[11]P/DB1[1 1]N	数字	I	0.825V~1 .575V	1.25GSPS /
L13,M13	DB1[12]P/DB1[1 2]N	数字	I	0.825V~1 .575V	1.25GSPS /
L14,M14	DB1[13]P/DB1[1 3]N	数字	I	0.825V~1 .575V	1.25GSPS /
N1,P1	DB0[0]P/DB0[0] N	数字	I	0.825V~1 .575V	1.25GSPS /
N2,P2	DB0[1]P/DB0[1] N	数字	I	0.825V~1 .575V	1.25GSPS /
N3,P3	DB0[2]P/DB0[2] N	数字	I	0.825V~1 .575V	1.25GSPS /
N4,P4	DB0[3]P/DB0[3] N	数字	I	0.825V~1 .575V	1.25GSPS /
N5,P5	DB0[4]P/DB0[4] N	数字	I	0.825V~1 .575V	1.25GSPS /

N6,P6	DB0[5]P/DB0[5] N	数字	I	0.825V~1 .575V	1.25GSPS /	
N7,P7	DB0[6]P/DB0[6] N	数字	I	0.825V~1 .575V	1.25GSPS /	
N8,P8	DB0[7]P/DB0[7] N	数字	I	0.825V~1 .575V	1.25GSPS /	
N9,P9	DB0[8]P/DB0[8] N	数字	I	0.825V~1 .575V	1.25GSPS /	
N10,P10	DB0[9]P/DB0[9] N	数字	I	0.825V~1 .575V	1.25GSPS /	
N11,P11	DB0[10]P/DB0[1 0]N	数字	I	0.825V~1 .575V	1.25GSPS /	
N12,P12	DB0[11]P/DB0[1 1]N	数字	I	0.825V~1 .575V	1.25GSPS /	
N13,P13	DB0[12]P/DB0[1 2]N	数字	I	0.825V~1 .575V	1.25GSPS /	
N14,P14	DB0[13]P/DB0[1 3]N	数字	I	0.825V~1 .575V	1.25GSPS /	

典型性能特征

普通模式输出

除非特殊说明, $I_{OUTFS} = 20\text{ mA}$, 标准电源, 25°C 。

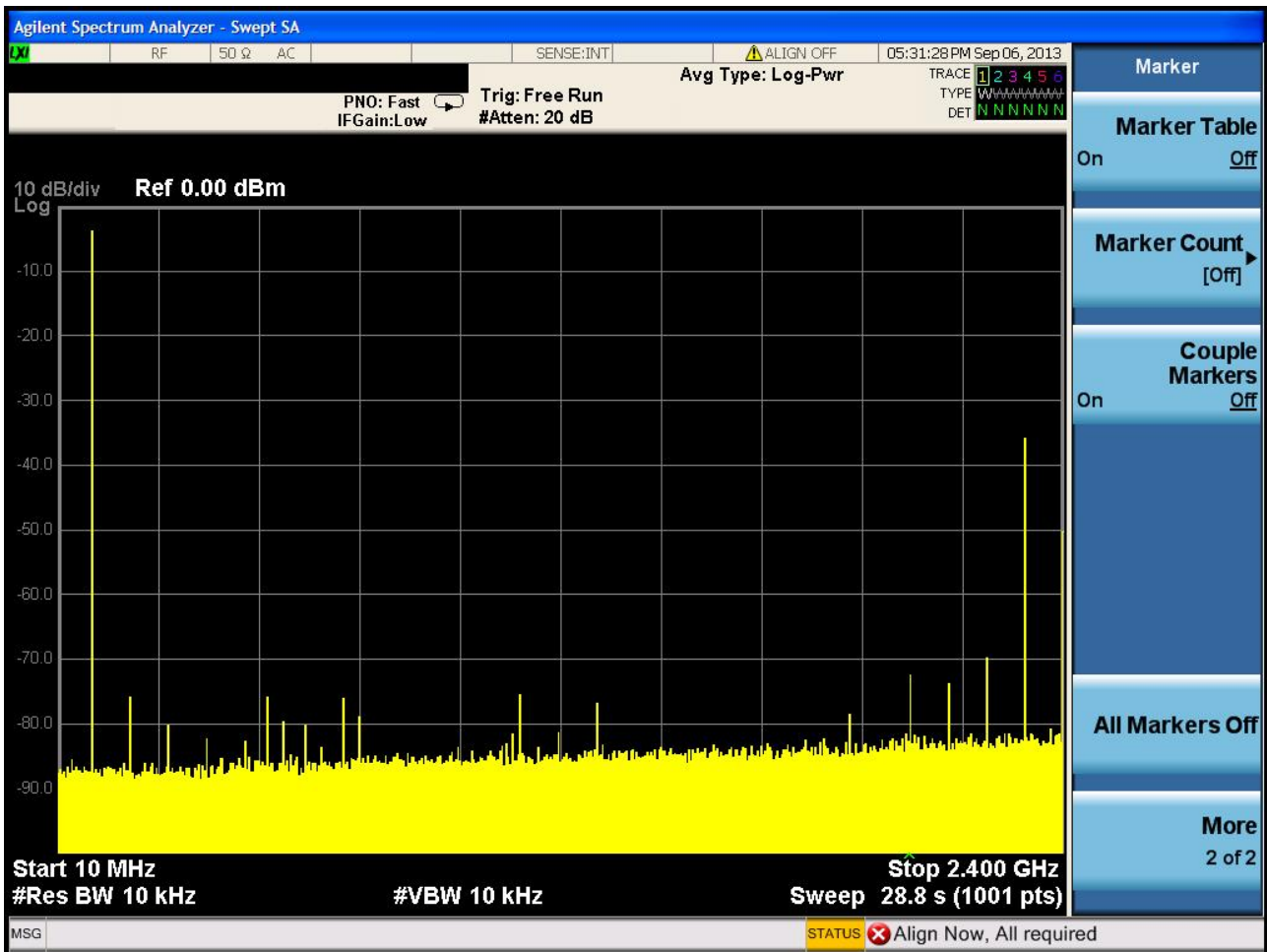


图 3 单音信号频谱, $f_{OUT}=91\text{MHz}$, $f_{DAC}=2.4\text{GSPS}$

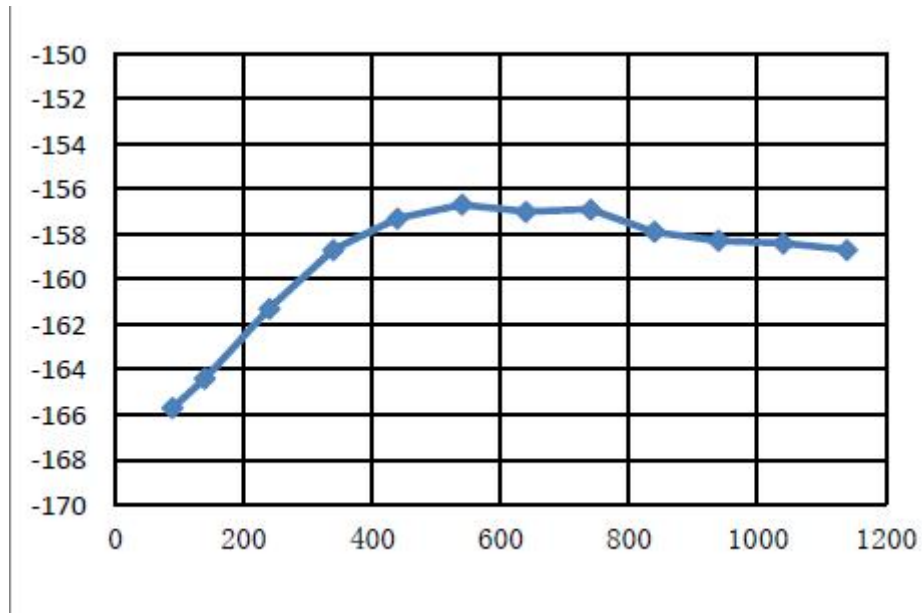


图 4 单音信号 NSD 随 f_{OUT} 变化曲线

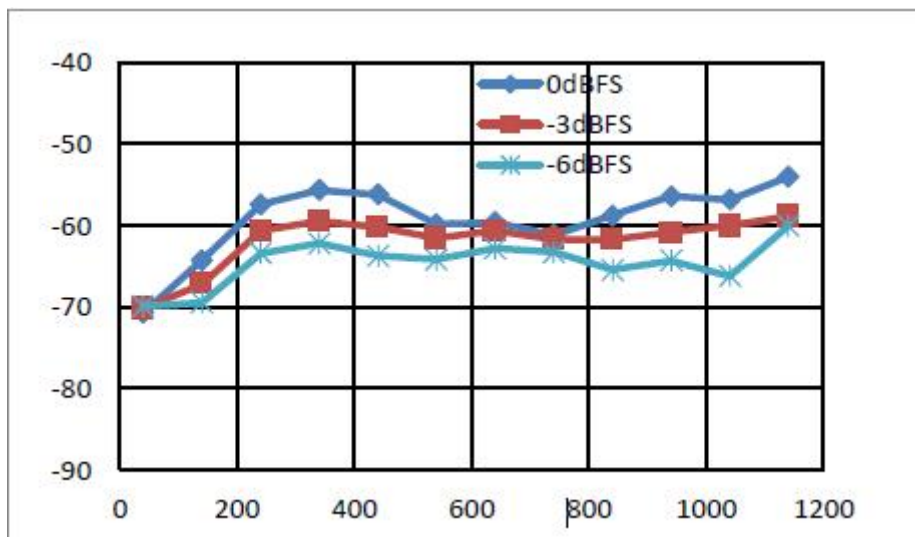


图 5 不同数字功率输入时 SFDR 随 f_{OUT} 变化曲线

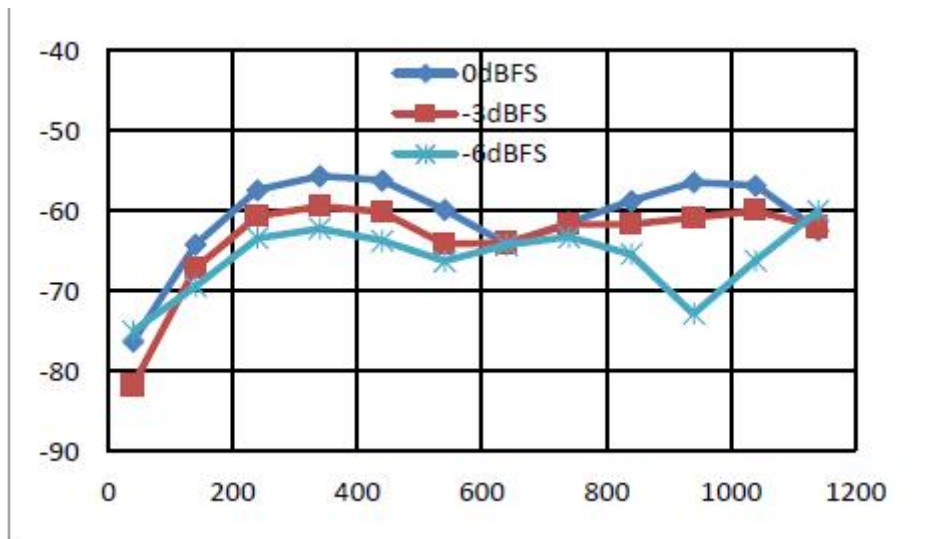


图 6 不同数字功率输入时二次谐波 SFDR 随 f_{OUT} 变化曲线

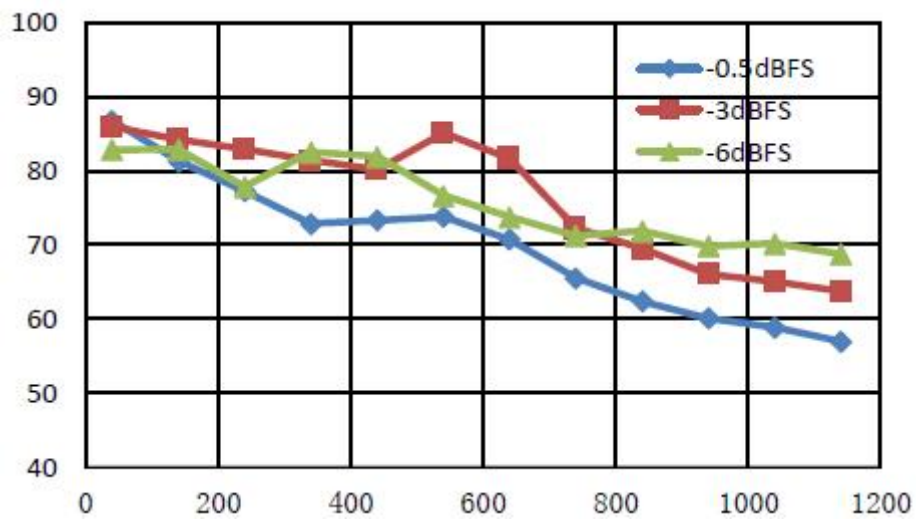


图 7 不同数字功率输入时 IMD 随 f_{OUT} 变化曲线

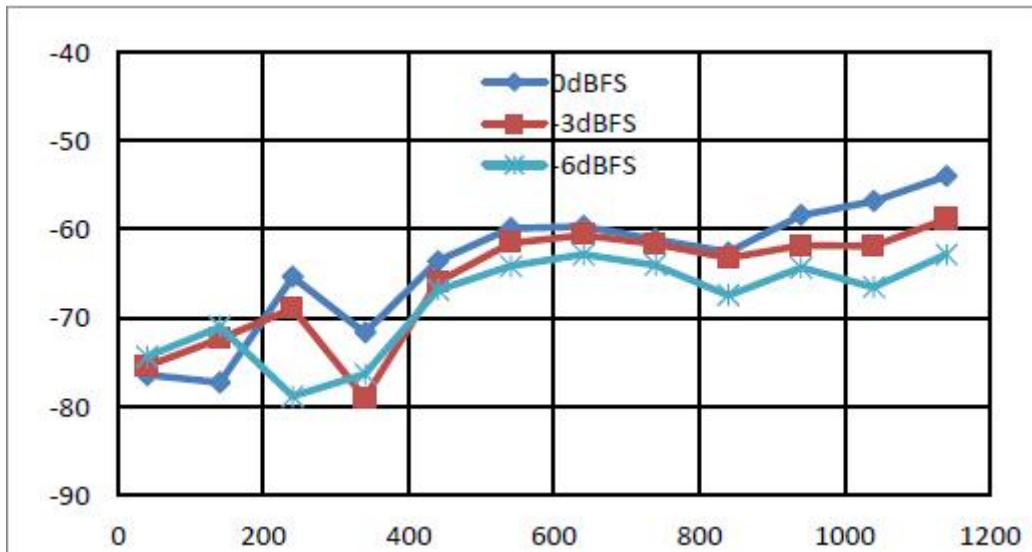


图 8 不同数字输入时三次谐波 SFDR 随 f_{OUT} 变化曲线

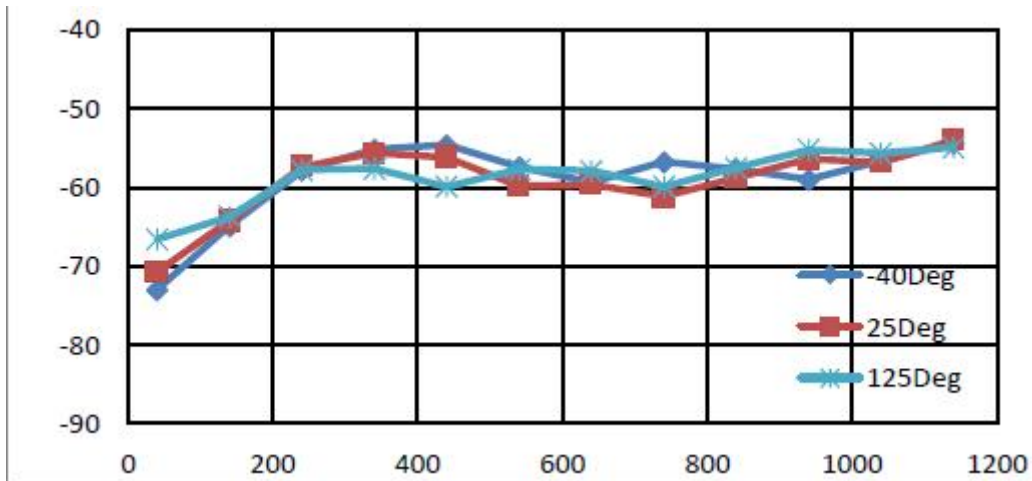


图 9 不同温度条件下 SFDR 随 f_{OUT} 变化曲线

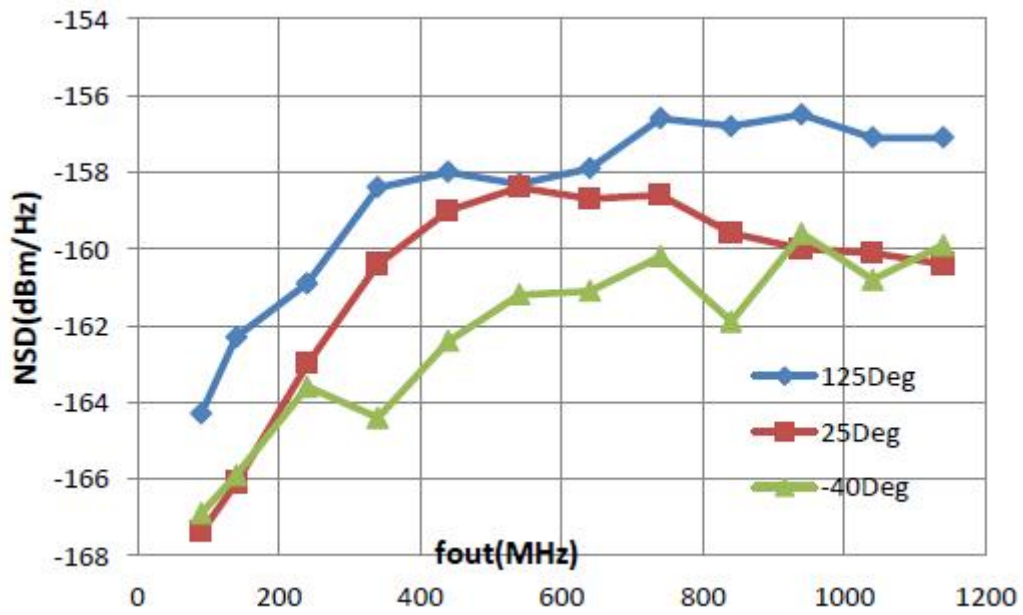


图 10 不同温度条件下 NSD 随 f_{OUT} 变化曲线

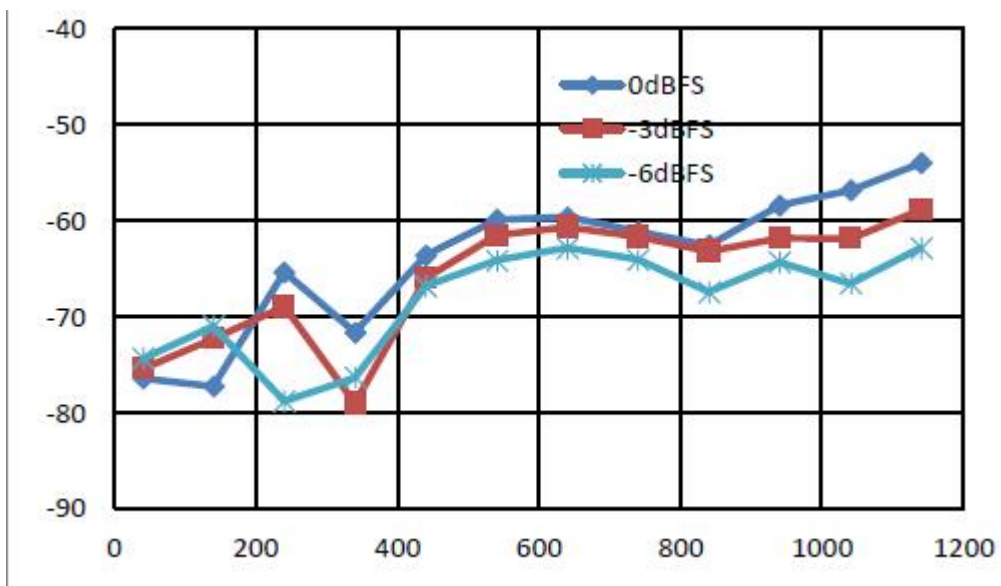


图 11 不同数字满量程时三次谐波 SFDR 随 f_{OUT} 变化曲线

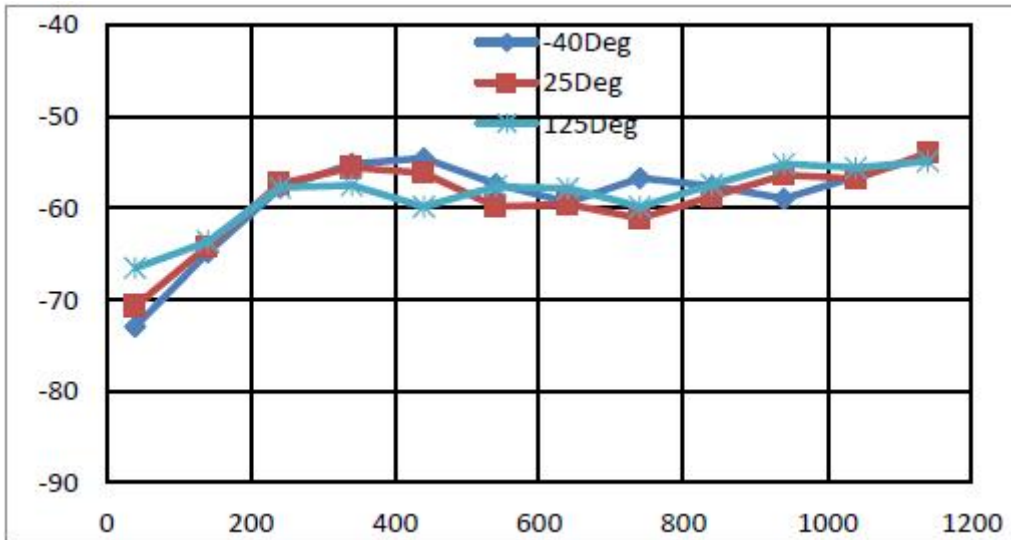


图 12 不同温度时单音信号 SFDR 随 f_{OUT} 变化曲线

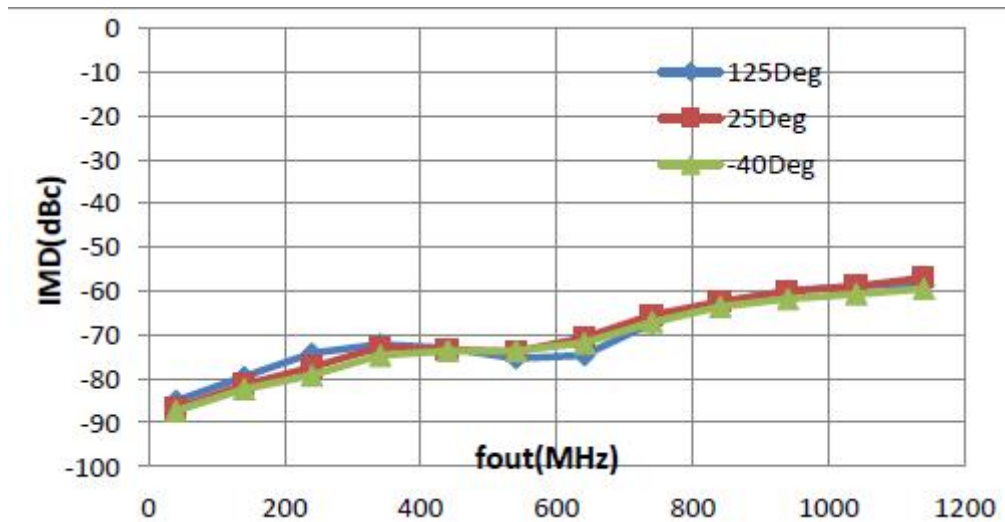


图 13 不同温度时 IMD 随 f_{OUT} 变化曲线

混频模式输出

除非特殊说明 $f_{DAC}=2.4\text{GSPS}$, $I_{OUTFS}=20\text{ mA}$, 标准电源, 25°C 。

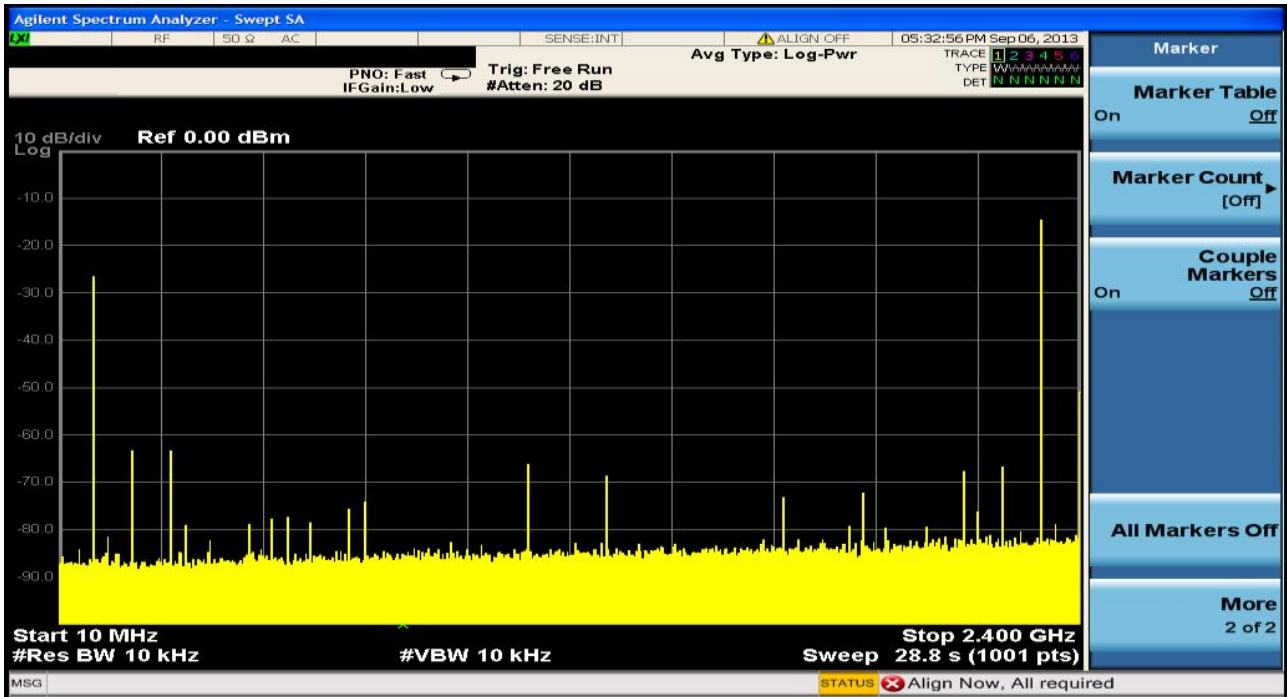


图 14 单音信号频谱, $f_{OUT}=2.31\text{GHz}$, $f_{DAC}=2.4\text{GSPS}$

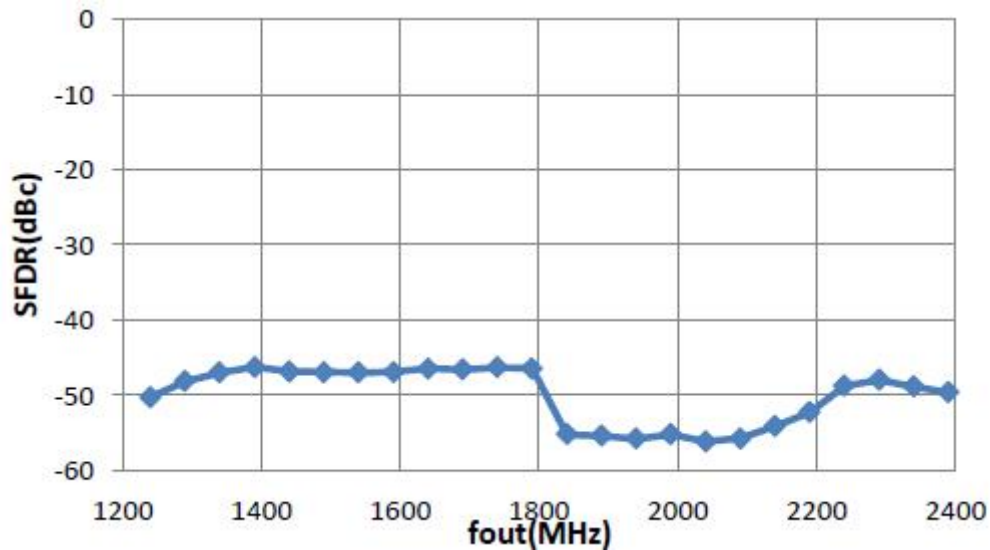


图 15 混频模式下 2.4GSPS 时, SFDR 随 f_{OUT} 变化曲线

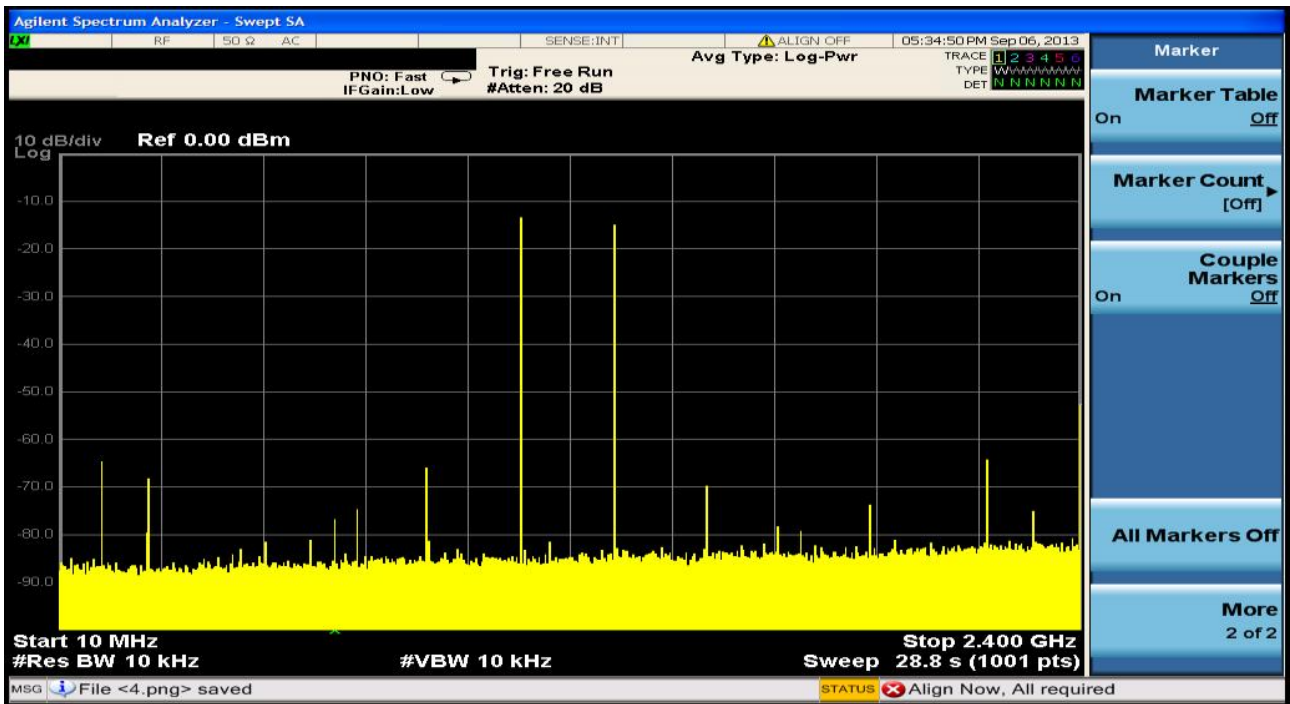


图 16 混频模式下单音信号频谱, $f_{OUT}=1.31\text{GHz}$, $f_{DAC}=2.4\text{GSPS}$

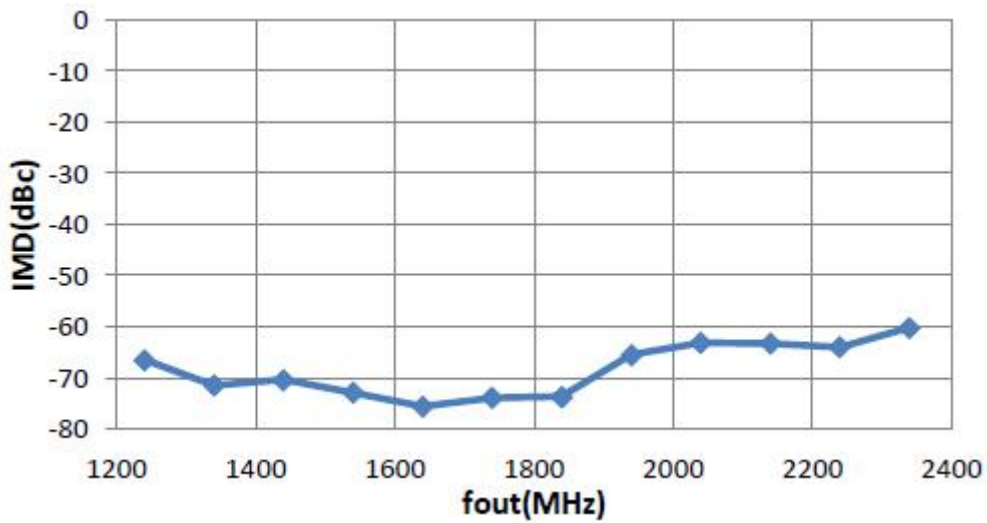


图 17 混频模式下 2.4GSPS 时 IMD 随 f_{OUT} 变化曲线

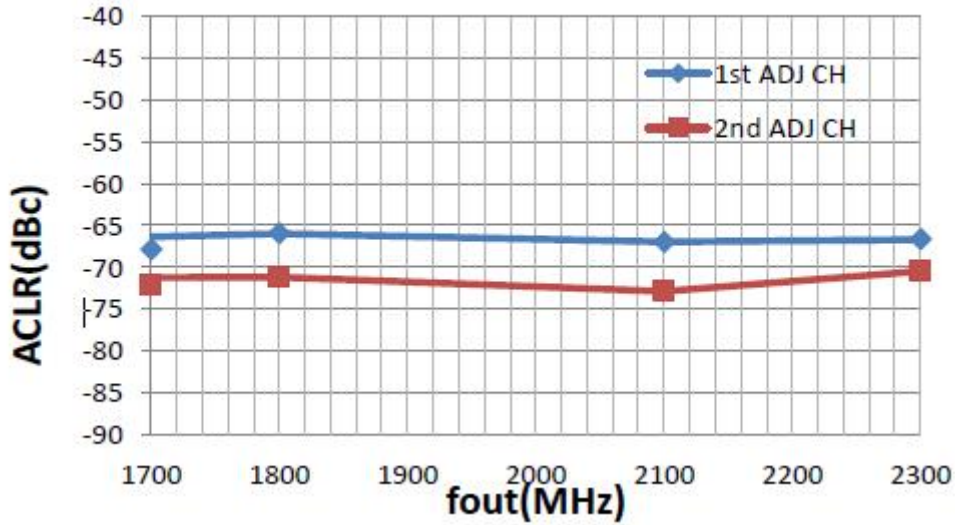


图 18 $f_{DAC}=2457\text{MSPS}$ 时，单音信号 WCDMA ACLR 随 f_{OUT} 变化曲线

串行控制 (SPI) 寄存器

CBM97D39BG 集成了可编程控制寄存器，用于控制器件的工作模式、状态等等一系列功能。有关寄存器的详细描述，请参考表 8。

复位

CBM97D39BG 内部有上电复位的功能，在使用过程中，推荐用户在上电后进行一次硬件或软件复位操作。软件复位通过寄存器 0x00 实现，有关软件复位的详细操作，请参考表 8 中有关该寄存器的详细描述。硬件复位是通过向 RESET 管脚（管脚 F14）加一个最小脉宽为 40ns 的高电平脉冲来实现。不用的时候，硬件复位管脚 RESET 需要连接到地。

SPI 操作

CBM97D39BG 的 SPI 包括串口时钟信号 (SCLK)，串口使能信号 (/CS)，串口数据输入/输出信号 (SDIO) 和串口数据输出信号 (SDO) 组成。串口兼容 3 线和 4 线模式，接口电平 3.3V。上电复位后，缺省状态为 4 线模式。可以通过设置 SDIO_DIR (寄存器 0x00，比特 7) 为 '1' 来实现 3 线 SPI 接口，这时 SDIO 为双向数据线，而 SDO 为高阻态。SCLK 的最大频率为 20MHz。

SPI 指令格式

表 7: SPI 指令格式

MSB				LSB			
17	16	15	14	13	12	11	10
R/W	A6	A5	A4	A3	A2	A1	A0

SPI 每次的读和写操作都有伴随着一个 8 位的指令头。最高位是 R/W 标识位, '1' 对应读操作, '0' 对应写操作, 位 6 ~ 位 0 描述数据传输时的寄存器地址。读和写操作时 8 比特数据都紧跟着指令头。对写操作, 在每个传输字节写入最后一位后寄存器立刻有效。片选信号 (/CS) 可以在每 8 位的序列 (除了最后一个字节) 后拉升为高电平来停止总线操作。/CS 当为低时, 串行传输继续。在非字节边界停止总线操作会复位 SPI 口。

CBM97D39BG 可以支持最高位 (MSB) 优先和最低位 (LSB) 优先两种数据传输方式, 其数据格式如图 19 所示, 可通过 SDIO_DIR 位 (寄存器 0x00 比特 7) 来选择, 上电复位后默认为 MSB 优先的数据格式。

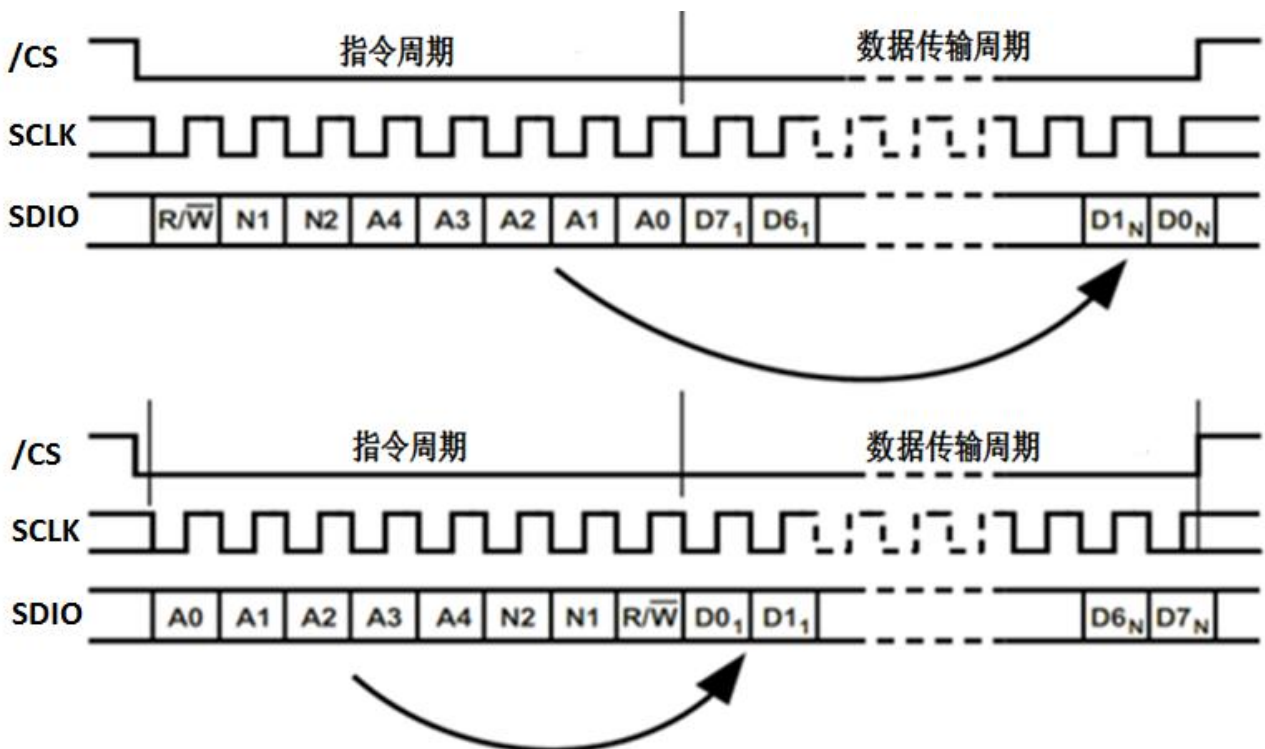


图 19 SPI 串口时序, MSB 优先 (上图) 和 LSB 优先 (下图)

SPI 端口的 3 线写操作的时序如图 20, 3 线读操作的时序如图 21, 4 线读写时序如图 22。

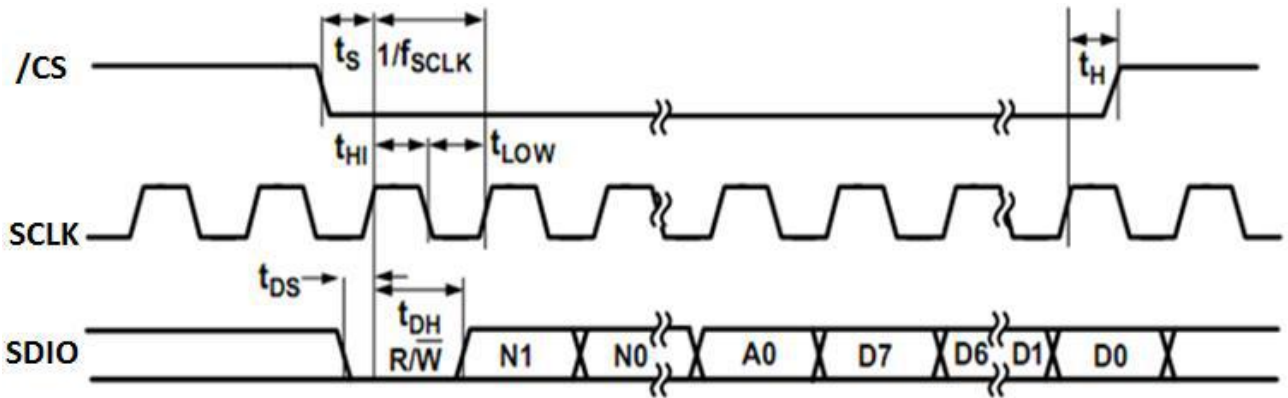


图 20 3 线 SPI 写操作时序

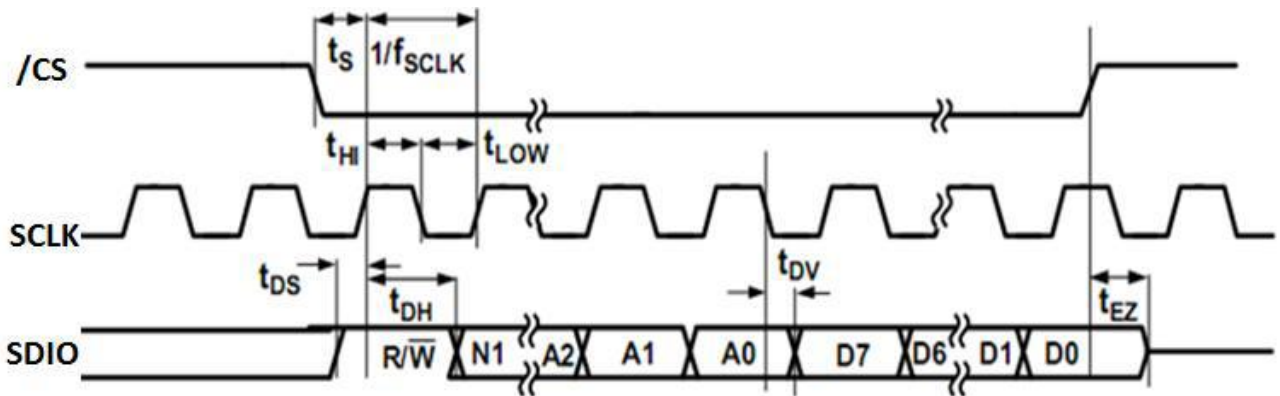


图 21 3 线 SPI 读操作时序

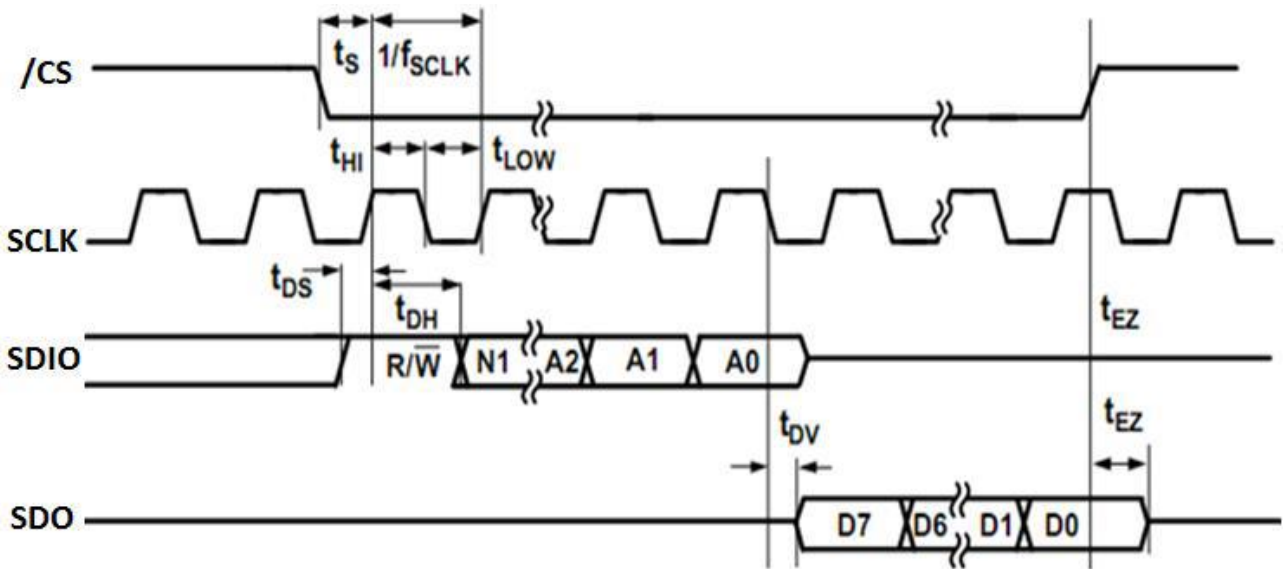


图 22 4 线 SPI 读操作时序

SPI 寄存器

表 8: 寄存器说明 (N/A=不可用)

名称	Hex 地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	缺省值
Mode	00	SDIO_DIR	LSB/MSB	Reset	N/A	N/A	N/A	N/A	N/A	0x00
Power-Down	01	N/A	N/A	LVDS_DRVR_PD	LVDS_RCVR_PD	N/A	N/A	CLK_RCVR_PD	DAC_BIAS_PD	0x00
CNT_CLK_DIS	02	N/A	N/A	N/A	N/A	CLKGEN_PD	N/A	REC_CNT_CLK	MU_CNT_CLK	0x03
IRQ_EN	03	N/A	N/A	SYNC_LST_EN	SYNC_LCK_EN	MU_LST_EN	MU_LCK_EN	RCV_LST_EN	RCV_LCK_EN	0x00
IRQ_REQ	04	N/A	N/A	SYNC_LST_IRQ	SYNC_LCK_IRQ	MU_LST_IRQ	MU_LCK_IRQ	RCV_LST_IRQ	RCV_LCK_IRQ	0x00
RSVD	05	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
FSC_1	06	FSC[7]	FSC[6]	FSC[5]	FSC[4]	FSC[3]	FSC[2]	FSC[1]	FSC[0]	0x00
FSC_2	07	Sleep	N/A	N/A	N/A	N/A	N/A	FSC[9]	FSC[8]	0x02
DEC_CNT	08	N/A	N/A	N/A	N/A	N/A	N/A	DAC_DEC[1]	DAC_DEC[0]	0x00
RSVD	09	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
LVDS_CNT	0A	LVDS_REG[7]	LVDS_REG[6]	N/A	N/A	HNDOFF_CHK_RST	N/A	LVDS_Bias[1]	LVDS_Bias[0]	0x00
DIG_STAT	0B	HNDOFF_Fall[3]	HNDOFF_Fall[2]	HNDOFF_Fall[1]	HNDOFF_Fall[0]	HNDOFF_Rise[3]	HNDOFF_Rise[2]	HNDOFF_Rise[1]	HNDOFF_Rise[0]	RNDM
LVDS_STAT1	0C	SUP/HLD_Edge1	N/A	DCI_PHS3	DCI_PHS1	DCI_PRE_PH2	DCI_PRE_PH0	DCI_PST_PH2	DCI_PST_PH0	RNDM
LVDS_STAT2	0D	SUP/HLD_SYNC	SUP/HLD_Edge0	SYNC_SAMP1	SYNC_SAMP0	LVDS1_HI	LVDS1_LO	LVDS0_HI	LVDS0_LO	RNDM/0
RSVD	0E	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
RSVD	0F	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

LVDS_REC_C NT1	10	SYNC_ FLG_RST	SYNC_ LOOP_O N	SYNC_ MST/SLV	SYNC_ CNT_EN A	N/A	RCVR_ FLG_RST	RCVR_ LOOP_O N	RCVR_ CNT_EN A	0x42
LVDS_REC_C NT2	11	SMP_DEL [1]	SMP_ DEL[0]	FINE_ DEL_ MID[3]	FINE_ DEL_ MID[2]	FINE_DEL _MID[1]	FINE_DEL _MID[0]	RCVR_ GAIN[1]	RCVR_ GAIN[0]	0xDD
LVDS_REC_C NT3	12	SMP_DEL [9]	SMP_ DEL[8]	SMP_ DEL[7]	SMP_ DEL[6]	SMP_ DEL[5]	SMP_ DEL[4]	SMP_ DEL[3]	SMP_ DEL[2]	0x29
LVDS_REC_C NT4	13	DCI_DEL[3]	DCI_ DEL[2]	DCI_ DEL[1]	DCI_ DEL[0]	FINE_DEL _SKW[3]	FINE_DEL _SKW[2]	FINE_DEL _SKW[1]	FINE_DEL _SKW[0]	0x71
LVDS_REC_C NT5	14	CLKDIVP H[1]	CLKDIVP H[0]	DCI_ DEL[9]	DCI_ DEL[8]	DCI_ DEL[7]	DCI_ DEL[6]	DCI_ DEL[5]	DCI_ DEL[4]	0x0A
LVDS_REC_C NT6	15	SYNC_ GAIN[1]	SYNC_ GAIN[0]	SYNCOU T_PH[1]	SYNCOU T_PH[0]	LCKTHR[3]	LCKTHR[2]	LCKTHR[1]	LCKTHR[0]	0x42
LVDS_REC_C NT7	16	N/A	SYNCO_ DEL[6]	SYNCO_ DEL[5]	SYNCO_ DEL[4]	SYNCO_ DEL[3]	SYNCO_ DEL[2]	SYNCO_ DEL[1]	SYNCO_ DEL[0]	0x00
LVDS_REC_C NT8	17	SYNCSH_ DEL[0]	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0x00
LVDS_REC_C NT9	18	SYNCSH_ DEL[8]	SYNCSH_ DEL[7]	SYNCSH_ DEL[6]	SYNCSH_ DEL[5]	SYNCSH_ DEL[4]	SYNCSH_ DEL[3]	SYNCSH_ DEL[2]	SYNCSH_ DEL[1]	0x00
LVDS_REC_ST AT1	19	SMP_DEL [1]	SMP_DEL [0]	N/A	N/A	SMP_ FINE_ DEL[3]	SMP_ FINE_ DEL[2]	SMP_ FINE_ DEL[1]	SMP_ FINE_ DEL[0]	0xC7
LVDS_REC_ST AT2	1A	SMP_DEL [9]	SMP_ DEL[8]	SMP_ DEL[7]	SMP_ DEL[6]	SMP_ DEL[5]	SMP_ DEL[4]	SMP_ DEL[3]	SMP_ DEL[2]	0x29
LVDS_REC_ST AT3	1B	DCI_DEL[1]	DCI_DEL[0]	N/A	N/A	SYNCOU T PH[1]	SYNCOU T PH[0]	CLKDIV PH[1]	CLKDIV PH[0]	0xC0
LVDS_REC_ST AT4	1C	DCI_DEL[9]	DCI_ DEL[8]	DCI_ DEL[7]	DCI_ DEL[6]	DCI_ DEL[5]	DCI_ DEL[4]	DCI_ DEL[3]	DCI_ DEL[2]	0x29
LVDS_REC_ST AT5	1D	FINE_DEL _PST[3]	FINE_DEL _PST[2]	FINE_DEL _PST[1]	FINE_DEL _PST[0]	FINE_DEL _PRE[3]	FINE_DEL _PRE[2]	FINE_DEL _PRE[1]	FINE_DEL _PRE[0]	0x86
LVDS_REC_ST AT6	1E	N/A	SYNCO_ DEL[6]	SYNCO_ DEL[5]	SYNCO_ DEL[4]	SYNCO_ DEL[3]	SYNCO_ DEL[2]	SYNCO_ DEL[1]	SYNCO_ DEL[0]	0x00

LVDS_REC_ST AT7	1F	SYNCSH_ DEL[0]	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0x00
LVDS_REC_ST AT8	20	SYNCSH_ DEL[8]	SYNCSH_ DEL[7]	SYNCSH_ DEL[6]	SYNCSH_ DEL[5]	SYNCSH_ DEL[4]	SYNCSH_ DEL[3]	SYNCSH_ DEL[2]	SYNCSH_ DEL[1]	0x00
LVDS_REC_ST AT9	21	SYNC_ TRK_ON	SYNC_ INIT_ON	SYNC_ LST_LCK	SYNC_LC K	RCVR_ TRK_ON	RCVR_ FE_ON	RCVR_LS T	RCVR_LC K	0x00
CROSS_CNT1	22	N/A	N/A	N/A	DUTY_EN	N/A	N/A	N/A	N/A	0x00
CROSS_CNT2	23	CROSS_C TRL[5]	N/A	N/A	CROSS_C TRL[4]	CROSS_C TRL[3]	CROSS_C TRL[2]	CROSS_C TRL[1]	CROSS_C TRL[0]	0x00
PHS_DET	24	N/A	N/A	CMP_BST	PHS_DET AUTO_E N	Bias[3]	Bias[2]	Bias[1]	Bias[0]	0x00
MU_DUTY	25	MU_ DUTYAU TO_EN	POS/NE G	ADJ[5]	ADJ[4]	ADJ[3]	ADJ[2]	ADJ[1]	ADJ[0]	0x00
MU_CNT1	26	N/A	Slope	Mode[1]	Mode[0]	Read	Gain[1]	Gain[0]	Enable	0x42
MU_CNT2	27	MUDEL[0]	SRCH_M ODE [1]	SRCH_M ODE[0]	SET_PHS[4]	SET_PHS[3]	SET_PHS[2]	SET_PHS[1]	SETPHS[0]	0x40
MU_CNT3	28	MUDEL[8]	MUDEL[7]	MUDEL[6]	MUDEL[5]	MUDEL[4]	MUDEL[3]	MUDEL[2]	MUDEL[1]	0x00
MU_CNT4	29	SEARCH_ TOL	Retry	CONTRS T	Guard[4]	Guard[3]	Guard[2]	Guard[1]	Guard[0]	0x0B
MU_STAT1	2A	N/A	N/A	N/A	N/A	N/A	N/A	MU_LOS T	MU_LKD	0x00
ANA_STAT	2B	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
RSVD	2C	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
ANA_CNT1	32	HDRM[7]	HDRM[6]	HDRM[5]	HDRM[4]	HDRM[3]	HDRM[2]	HDRM[1]	HDRM[0]	0xCA
ANA_CNT2	33	N/A	N/A	N/A	N/A	N/A	N/A	MSEL[1]	MSEL[0]	0x03
RSVD	34	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
PART ID	35	ID[7]	ID[6]	ID[5]	ID[4]	ID[3]	ID[2]	ID[1]	ID[0]	0x20

SPI 端口配置和软件复位寄存器

表 9: SPI 端口配置和软件复位寄存器

地址 (十六进制)	名称	位	R/W	默认值	注释
0x00	SDIO_DIR	7	R/W	0	
	LSB/MSB	6	R/W	0	
	Reset	5	R/W	0	

LVDS 接口和 DAC 断电控制寄存器

表 10: LVDS 接口和 DAC 断电控制寄存器

地址 (十六进制)	名称	位	R/W	默认值	注释
0x01	LVDS_DRVR_PD	5	R/W	0	LVDS驱动/接收机和DAC的断电。0=使能, 1=关闭。
	LVDS_RCVR_PD	4	R/W	0	
	CLK_PCVR_PD	1	R/W	0	
	DAC_BIAS_PD	0	R/W	0	

控制器时钟使能寄存器

表 11: 内部时钟使能寄存器

地址 (十六进制)	名称	位	R/W	默认值	注释
0x02	CLKGEN_PD	3	R/W	0	内部时钟分配使能 0=使能, 1=关闭。
	REC_CNT_C LK	1	R/W	1	LVDS接收机和Mu控制器时钟使能 0=关闭, 1=使能
	MU_CNT_C LK	0	R/W	1	

中断请求 (IRQ) 使能/状态

表 12: 中断请求 (IRQ) 使能/状态寄存器

地址 (十六进制)	名称	位	R/W	默认值	注释
0x03	SYNC_LST_EN	5	W	0	该寄存器使sync, mu和LVDS Rx控制器可以更新它们在寄存器0x04中对应的IRQ状态比特位, 寄存器0x04定义了是锁定状态 (LCK) 还是失锁状态 (LST)。 0=关闭 (复位状态位), 1=使能。
	SYNC_LCK_EN	4	W	0	
	MU_LST_EN	3	W	0	
	MU_LCK_EN	2	W	0	
	RCV_LST_EN	1	W	0	
	RCV_LCK_EN	0	W	0	
0x04	SYNC_LST_IRQ	5	R	0	该寄存器指示控制器的状态。对LCK_IRQ比特: 0=未失锁, 1=锁定。对LST_IRQ比特: 0=未失锁, 1=未锁定。注意, 如果控制器IQR功能使能, 则寄存器0x03中的相对应的中断控制位需要先写0来复位, 然后写1来使能。
	SYNC_LCK_IRQ	4	R	0	
	MU_LST_IRQ	3	R	0	
	MU_LCK_IRQ	2	R	0	
	RCV_LST_IRQ	1	R	0	
	RCV_LCK_IRQ	0	R	0	

DAC 满量程电流设置 (I_{OUTFS}) 和睡眠控制寄存器

表 13: DAC 满量程电流设置 (I_{OUTFS}) 和睡眠控制寄存器

地址 (十六进制)	名称	位	R/W	默认值	注释
0x06	FSC_1	[7:0]	R/W	0	V设置DAC的 I_{OUTFS} 在8mA到31mA之间 (默认值为20mA)。 $I_{OUTFS}=0.0226 \times FSC[9:0]+8.58, FSC=0 \sim 1023$
0x07	FSC_2	[1:0]	R/W	0x02	
	Sleep	7	R/W		0= DAC 输出使能, 1=DAC 输出关闭 (睡眠)

DAC 四项-开关工作模式寄存器

表 14: DAC 四相-开关工作模式寄存器

地址 (十六进制)	名称	位	R/W	默认值	注释
0x08	DAC-DEC	[1:0]	R/W	0x00	0x00=普通基带模式 0x01=归零模式 0x02=混频模式

DCI 相位调整状态

表 15: DCI 相位调整状态寄存器

地址 (十六进制)	名称	位	R/W	默认值	注释
0x0C	DCI_PRE_PHO	2	R	0	0=DCI上升在相位0的PRE延迟信号采样沿之后, 1=DCI上升沿在相位0的PRE延迟信号采样沿之前
	DCI_PST_PHO	0	R	0	0=DCI上升沿在相位0的POST延迟信号采样沿之后, 1=0=DCI上升沿在相位0的POST延迟信号采样沿之前

SYNC_IN 相位调整状态

表 16: SYNC_IN 相位调整状态寄存器

地址 (十六进制)	名称	位	R/W	默认值	注释
0x0D	SYNC_IN_PH9 0	5	R	0	0=SYNCIN上升沿在Phase90采样沿之后 1=SYNCIN上升沿在Phase90采样沿之前
	SYNC_IN_PHO	4	R	0	0=SYNCIN上升沿在Phase0采样沿之后 1=SYNCINI上升沿在Phase0采样沿之前

数据接收控制寄存器配置

表 17: 数据接收控制寄存器配置

地址 (十六进制)	名称	位	R/W	默认值	注释
0x10	SYNC_FLG_RST	7	W	0	Sync控制器标志位复位。写1后再写0复位标志位。
	SYNC_LOOP_ON	6	R/W	1	0=关闭, 1=使能。只有主触发器的使能。使能时, 主触发器不再锁住, 当sync控制器产生一个IRQ时, 便自动开始搜索/跟踪流程。
	SYNC__MST/	5	R/W	0	Sync控制器配置。0=从触发器, 1=主触发器。

	SLV				
	SYNC_CNT_ENA	4	R/W	0	Sync 控制器使能。0=关闭, 1=使能。
	RCVR_FLG_RST	2	W	0	数据接收机控制器标志位复位。写1后再写0复位标志位。
	RCVR_LOOP_ON	1	R/W	1	0=关闭, 1=使能。使能时, 当数据接收机控制器产生一个 IRQ 中断时, 不再锁住, 自动开始搜索/跟踪流程。
	RCVR_CNT_ENA	0	R/W	0	数据接收机控制器使能。0=关闭, 1=使能。

数据接收控制器_数据采样延时值寄存器

表 18: 数据接收机控制器_数据采样延时值寄存器

地址 (十六进制)	名称	位	R/W	默认值	注释
0x11	SMP_DEL[1:0]	[7:6]	R/W	11	控制器使能: 10 比特值 (最大值 332) 代表延迟线的起始值, 该值被状态机用来对输入数据进行采样。缺省值为 167, 表示位于延迟线的中点。控制器关闭时, 该寄存器为实际延迟线的值。
0x12	SMP_DEL[9:2]	[7:0]	R/W	0x25	

数据和 SYNC 接收机控制器_DCI 延时值/窗口和相位切换寄存器

表 19: 数据和 SYNC 接收机控制器_DCI 延时值/窗口和相位切换寄存器

地址 (十六进制)	名称	位	R/W	默认值	注释
0x13	DCI_DEL[3:0]	[7:4]	R/W	0111	参考寄存器 0x14 的 DCI_DEL 描述。
	FINE_DEL_SKEW	[3:0]	R/W	0001	4 bit 的值设置 DCI PRE 和 POST 采样时钟的差值 (窗口)。默认值 1 为窄窗口。
0x14	CLKDIVPH[1:0]	[7:6]	R/W	00	内部除 4 电路的相关相位。这个特性允许以 90° 为增量 (每加 1) 的相位切换, 拓展 Rx 控制器的时钟锁存范围从 0.8GPSP 到 1.6GPSP (只有在 sync 控制器关闭时才有效)。
	DCI_DEL[9:4]	[5:0]	R/W	001010	控制器使能时: 10 比特值 (最大为 332) 表示采样 DCI 输入的状态机的延迟线的初始值。默认值 167 表示为延迟线的中点。控制器关闭时: 寄存器的值为延

					迟线的真实值。
0x15	SYNC GAIN[1:0]	[7:6]	R/W	00	设置 sync 跟踪增益（最优值为 1）。
	SYNCOOUT_PH[1:0]	[5:4]	R/W	00	当前 SYNC_OUT 相位选择的读回值。
	LCKTHR[3:0]	[3:0]	R/W	0000	设置采样和 DCI 延时之间的锁定差值（最优值为 2）。
0x16	SYNCO_DEL[6:0]	[6:0]	R/W	0x00	当 sync 控制器关闭时设置 sync 输出延迟值；反之，当 sync 使能时，代表的是 sync 输出延迟值的读取状态。
0x17	SYNCO_DEL[0]	[7]	R/W	0x00	当 sync 控制器关闭时设置 sync 的建立和保持延迟值；反之，当 sync 使能时，代表的是 sync 建立和保持值的读取状态。
0x18	SYNCO_DEL[8:1]	[7:0]	R/W	0x00	当 sync 控制器关闭时设置 sync 的建立和保持延迟值；反之，当 sync 使能时，代表的是 sync 建立和保持值的读取状态。

数据接收机控制器_延迟线状态和 SYNC 控制器 SYNC_OUT 状态寄存器

表 20: 数据接收机控制器_延迟线状态和 SYNC 控制器 SYNC_OUT 状态寄存器

地址 (十六进制)	名称	位	R/W	默认值	注释
0x19	SMP_DEL[1:0]	[7:6]	R	00	寄存器中数据表示的是在 DCI 和状态机完成搜索进入跟踪态后由数据接收控制器（使能时）确定的数据延迟线的真实值。注意这些值应该相等。SYNCOOUT_PH 提供相位多选器的相位状态（0/90/180/270），CLKDIVPH 提供数据接收控制器（寄存器 0x14）的相位状态。
0x1A	SMP_DEL[9:2]	[7:0]	R	0x00	
0x1B	SYNCOOUT_PH[1:0]	3:2	R	00	
	CLKDIV PH[1:0]	1:0	R	00	
	DCI_DEL[1:0]	[7:6]	R	00	
0x1C	DCI_DEL[9:2]	[7:0]	R	0x00	

SYNC 和数据接收机控制器锁定/跟踪状态寄存器

表 21: SYNC 和数据接收机控制器锁定/跟踪状态寄存器

地址 (十六进制)	名称	位	R/W	默认值	注释
0x21	SYNC_TRK_ON	7	R	0	SYNC_TRK_ON, RCVR_TRK_ON: 0=跟踪未建立, 1=跟踪已建立。 SYNC_LCK, RCVR_LCK: 0=控制器未锁定, 1=控制器已锁定。 SYNC_LST, RCVR_LST: 0=未失锁, 1=在某点已失锁。
	SYNC_LST	5	R	0	
	SYNC_LCK	4	R	0	
	RCVR_TRK_ON	3	R	0	
	RCVR_LST	1	R	0	
	RCVR_LCK	0	R	0	

CLK 接收机控制寄存器

表 22: CLK 接收机控制寄存器

地址 (十六进制)	名称	位	R/W	默认值	注释
0x22	DUTY_EN	4	R/W	0	1=时钟占空比控制环路工作 (推荐) 0=时钟占空比控制环路不工作
0x23	CROSS_AUTO_B	7	R/W	0	1=时钟交叉点控制在没有检测到信号时也工作 0=时钟交叉点控制在检测到信号时才工作 (推荐)
	CROSS_ON	4	R/W	0	1=时钟交叉点控制环路工作 (推荐) 0=时钟交叉点控制环路不工作
	CROSS_DIR	3	R/W	0	1=时钟交叉点调低 (推荐) 0=时钟交叉点调高
	CROSS_OFFSET	[2:0]	R/W	000	111=时钟交叉点调整幅度最大 (推荐) 000=时钟交叉点调整幅度最小

MU 控制器配置和状态寄存器

表 23: MU 控制器配置和状态寄存器

地址 (十六进制)	名称	位	R/W	默认值	注释
0x24	CMP_BST	5	R/W	0	相位检测器使能和偏置提升的控制位。建议设为 1 以使能该功能。
	PHS_DET AUTO_EN	4	R/W	0	
0x25	MU_DUTY AUTO_EN	7	R/W	0	Mu 控制器占空比调整使能位。建议设为 1 使能该功能。
0x26	Slope	6	R/W	1	Mu 控制器相位坡度锁定。0=负的坡度, 1=正的坡度。最优设定参考表 28。
	Mode[1:0]	[5:4]	R/W	00	设置 mu 控制器的工作模式。00=搜索并跟踪 (推荐模式), 01=只搜索, 10=跟踪。
	Read	3	R/W	0	设置为 1 时读取 Mu 延迟线的当前值。
	Gain[1:0]	[2:1]	R/W	01	设置 mu 控制器的跟踪增益。推荐使用默认的 01 值。
	Enable	0	R/W	0	1=使能 mu 控制器, 0=关闭 mu 控制器。
0x27	MUDEL[0]	7	R/W	0	9 比特 MUDEL 的最低位
	SRCH_MODE[1:0]	[6:5]	R/W	0	设置 mu 控制器搜索最优 mu 延迟线设置的初始方向 (从初始 MUDEL 设置开始), 该最优 mu 延迟线设置与期望的相位/坡度设置 (即 SET_PHS 和 Slope) 相关。00=向下, 01=向上, 10=向下/向上 (推荐设置)。
	SET_PHS[4:0]	[4:0]	R/W	0	设置 mu 控制器锁定的目标相位, 最大值为 16。最优优化设置参考表 28。
0x28	MUDEL[8:1]	[7:0]	W	0x00	当使能 (寄存器 0x26 的比特 0) 设置为 0 时, 该 9 比特值代表 mu 延迟的值。注意最大值为 432。当使能设置为 1 时, 该值代表控制器开始搜索的 mu 延迟值。推荐将这个值设为延迟线中点 216。
			R	0x00	当读取 (寄存器 0x26 比特 3) 设为 1 时, 使能为 0 时读回值为写入寄存器的值, 使能为 1 时读回值为 mu 控制器锁定的值。

0x29	SEARCH_TOL	7	R/W	0	不精确（可以在期望的相位两个值中间找到一个相位），1=找到期望的精确相位（最优设置）。
	Retry	6	R/W	0	0=如果找不到正确值则停止，1=如果找不到正确值则重试。
	CONTRST	5	R/W	0	当找不到期望相位时，控制控制器是复位还是继续。 0=继续（最优设置），1=复位。
	Guard[4:0]	4	R/W	01011	在 mu 延迟线的开始和结尾间设置警戒带，只有在警戒带外找不到可用相位时才进入警戒带（最优值是十进制 11 或 0x0B）。
0x2A	MU_LST	1	R	0	0=mu 控制器未失锁， 1=mu 控制器失锁。
	MU_LKD	0	R	0	0=mu 控制器未锁定， 1=mu 控制器锁定。

器件标识寄存器

表 24 器件标识寄存器

地址（十六进制）	名称	位	R/W	默认值	注释
0x35	PART_ID	[7:0]	R	0x20	器件识别号

工作说明

LVDS 数据接收

CBM97D39BG 采用双路 14 位 LVDS 的 DDR 数据接口 DB0 和 DB1，支持的总输入数据率为 1.6GSPS 到 2.5GSPS。

接口信号必须保证数据上升、下降沿对齐，且主机需提供与数据同步的频率为 FDAC/4 数据时钟输入（DCI）。数据采用上升、下降双沿传输（DDR 模式），数据格式为偏移二进制格式。图 23 为处理器与 CBM97D3 之间的接口示意图。

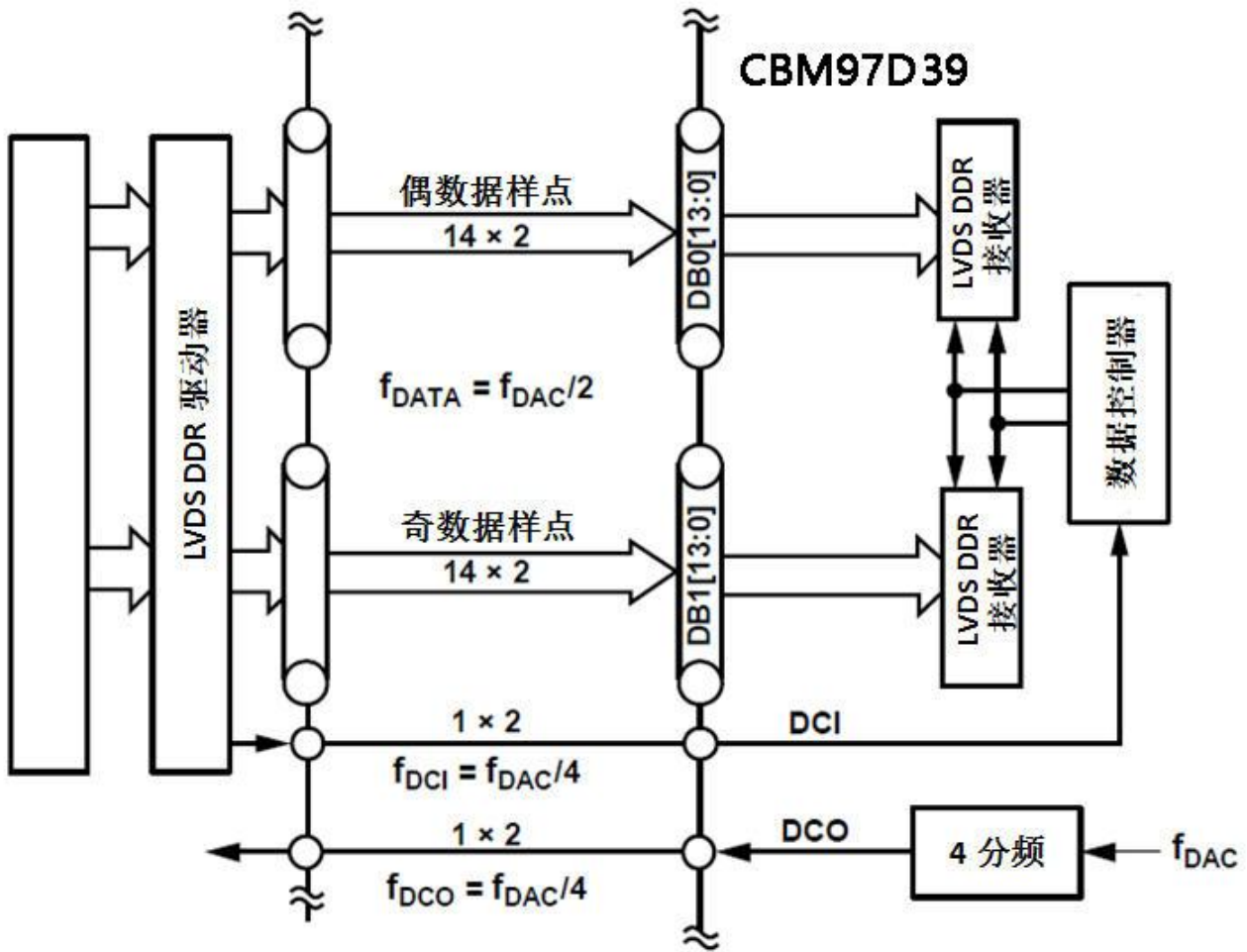


图 23 外部处理器与 CBM97D39BG 的接口示意图

CBM97D39BG 内部数据接收器产生一个与 DCI 相移 90°的时钟对输入数据进行采样。图 24 为推荐的接口时序。DCI 时钟和输入数据一样，必须保证尽可能低的抖动、偏移和码间干扰。为了保证 DCI 和数据的一致性，主机需要产生一个 010101...的数据当作 DCI 信号。为了尽可能保证数据接收机的稳定性，DCI 和数据的眼图窗口必须充分张开。为了降低各位之间的延迟失配，在 PCB 电路设计时需要保证每一位的差分阻抗匹配。

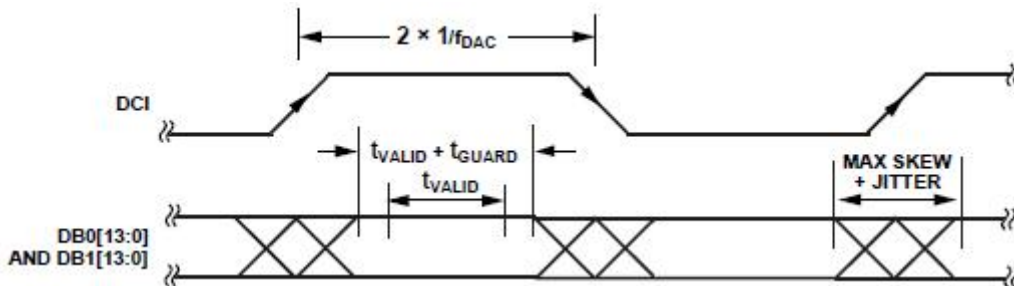


图 24 LVDS 数据口定时需求

所有数据相对 DCI 所允许的最大偏差和抖动如下:

$$\begin{aligned} \text{MaxSkew} + \text{Jitter} &= \text{Period}(\text{ns}) - \text{ValidWindow}(\text{ps}) - \text{Guard} \\ &= 800 \text{ ps} - 344 \text{ ps} - 100 \text{ ps} \\ &= 356 \text{ ps} \end{aligned}$$

在图 24 中 t_{VALID} 代表 ValidWindow (ps) , t_{GUARD} 代表 Guard。最小的 LVDS 有效窗为 344ps, 防护区推荐为 100ps。因此,在最大工作频率 2.5GSPS 下,FPGA 和 PCB 上最大允许的位偏差加抖动为 356ps。

为方便实现与主处理器之间的同步,CBM97D39BG 提供了一个和 DCI 同频 ($F_{\text{DAC}}/4$) 的数据时钟输出 DCO。DCO 由单独的时钟分频器产生,它与数据接收器收到的 $F_{\text{DAC}}/4$ 时钟之间的相位关系在每次上电后都是不确定的,这会有 ± 2 个 DACCLK 延迟偏差,因此在敏感场合下必须考虑同步控制。

主机的 DCO 和 DCI 之间的最大偏差受到制造工艺和实现的影响,另外,由于温度和供电电压的变化也会带来最高到 30%的偏差。当数据接收机中的延时链锁定后,其可容忍的延时差为 $\pm 1.5\text{ns}$ 。

图 25 为 CBM97D39BG 内部的延迟锁定环结构,用来保证主机和 CBM97D39BG 之间的数据传递。电路产生一个与 DCI 差 90° 的时钟来采样数据,从而保证采样时钟处于数据中央(假设 DCI 和 DBx[13:0]

延迟匹配)。

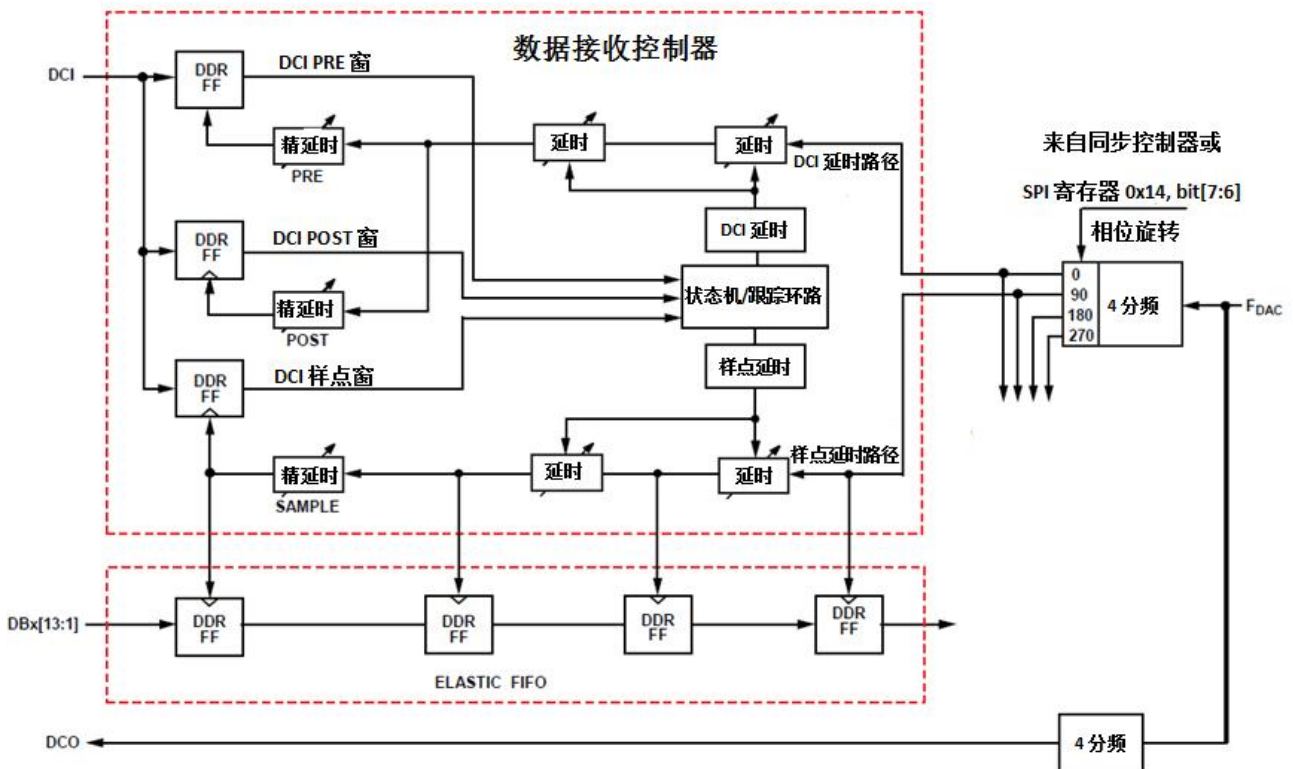


图 25 数据接收控制框图

数据接收完成后, CBM97D39BG 利用 FIFO 来将数据传递到 DAC。为了跟踪两个时钟域之间相位变化, 需要将数据接收器设置到跟踪模式 (寄存器 0X10, 位 1 和位 0)。跟踪模式下通过两个内部时钟 (PRE 和 PST)

在一个很窄的窗口内对 DCI 采样, 保证采样时钟跟踪 DCI 的变化, 如图 26 所示。

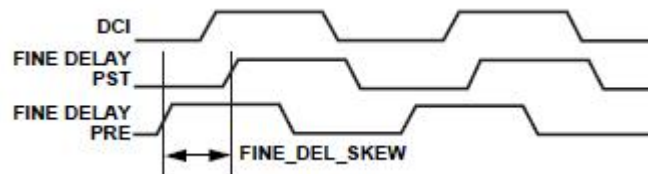


图 26Pre 延时时钟和 Pst 延时时钟采样图

对 DCI 的采样结果可以通过读取 DCI_PRE_PH0 (寄存器 0X0C, 位 2) 和 DCI_PST_PH0 (寄存器 0X0C, 位 0) 的值得到。如果延迟设置正确, DCI_PRE_PH0 应为 0, DCI_PST_PH0 应为 1。这些状态可能会因为时钟

抖动超出窗口宽度而发生瞬间切换, 因而控制器平均了多次状态结果来避免这个问题。窗口宽度 (FINE_DEL_SKEW) 通过寄存器 0x13 (位[3:0]) 设置, 其最大值 180ps, 分辨率为 12ps。推荐初始时偏差设置为 36ps (寄存器 0x13=0x72)。偏差的设定同样影响着控制环路的速度, 更严格的控制会导致更长的响应时间。

数据接收机初始化

上电复位后, 需要将数据接收机通过 SPI 初始化并设置为跟踪模式, 推荐初始化步骤如下:

1. 设定 FINE_DEL_SKEW 为 2, 以获得更大的 DCI 采样窗口 (寄存器 0x13=0x72)。默认的 DCI_DEL 和 SMP_DEL 设定 167 为最优值;
2. 芯片使能前先关闭控制器 (寄存器 0x10=0x00);
3. 分两步使能 Rx 控制器: 先设置寄存器 0x10=0x02, 然后设置寄存器 0x10=0x03;
4. 等待 135000 个时钟周期;
5. 读回寄存器 0x21, 并确认等于 0x05, 从而保证 DLL 环路锁定和跟踪;
6. 在低于 1.6GSPS 时钟的条件下, 需要执行该步骤。读回 DCI_DEL 的值, 确定其处于用户定义的跟踪保护带内。如果不是, 将 CLKDIVPH 设为 1 (寄存器 0x1, 位[7:6]), 回到步骤 2。

一旦在 SPI 初始化中开启数据接收机控制器, 控制器就进入搜索模式, 通过自动调整采样 DCI 和数据输入的时钟的延迟, 来寻找与 DCI 时钟最近的上升沿 (相对内部 $F_{DAC}/4$ 时钟)。状态机从 DCI_DEL 初始值开始搜索。首先搜索 DCI_DEL 上方第一个上升沿, 再搜索 DCI_DEL 下方第一个上升沿。状态机从中选择一个最近的上升沿, 并进入跟踪模式。建议使用 DCI_DEL 和 SMP_DEL 默认的中间刻度延迟值 (十进制 167), 这样能保证选择的边沿始终最接近延迟线的中点, 从而能提供更大范围的搜索延时调节量并避免控制器失

锁。内部时钟 (DCI 和延时后的采样时钟) 可调整的延迟范围一般为 4ns。10 位 (二进制) 的延迟控制字 (寄存器 0x10、0x14) 可调值的范围为 0 到 384 (十进制码), 分辨率大约为 12ps/LSB。当控制器开启后, 它会更新这些寄存器。当前延迟值与最大/最小值 (0 和 334) 之间差值的最小值为跟踪保护带。因此如果控制器稳定在 DCI_DEL=80 和 SMP_DEL=304, 则控制器可以有 80 个码的保护带 (大约 1ns) 来跟踪时钟域之间的相位变化。

CBM97D39BG 初始后数据接收控制器需要一定时间来锁定 DCI 信号。由于受到 mu 控制器和同步控制器 (可选) 的影响, 需要等到这些控制器的输出信号稳定后才能将数据接收控制器开启。所有的控制器都工作在 DAC 主时钟的分频时钟上。锁定 DCI 信号所需要的 DAC 时钟周期数依赖于同步控制器是否开启, 具体参见表 25。

表 25: LVDS 控制器典型/最差情况下的锁定时间 (相对于 $1/f_{DAC}$)

同步控制器	典型值	最差值
Off (关)	70K	135K
Slave (从模式)	70K	135K
Master (主模式)	300K	560K

在 SPI 初始过程中, 用户可以通过读取寄存器 0x21 (位 0, 位 1 和位 3) 的状态来判断数据接收控制器是锁定、失锁、还是进入跟踪模式。或者, 也可以通过使用 IRQ 位 (寄存器 0x03 和寄存器 0x04), 让控制器锁定后发出中断信号。只有在确定数据接收控制器锁定后, 才能开启数据接收机, 接收外部数输入。

数据接收控制器可以配置为失锁中断请求 (IRQ) 使能模式。失锁可能会由于 DCI 抖动过大、DAC 主时钟突变、或电源电压变化导致。为了响应中断, 主机可以通过查询 RCVR_CLK 位 (寄存器 0x21, 位 0) 的值来确定控制器目前的状态。如果该位为零, 可以把寄存器 0x10 中位 1 (RCVR_LOOP_ON 位) 置成 1 来重启搜索/跟踪过程。在等待所需的时间后, 主机可以通过 RCVR_CLK 位来判断是否完成。退出中断时, 需要将 RCVR_FLG_RST 位 (寄存器 0x10, 位 2) 清除, 方法是将该位先写 1 然后再写 0。

数据接收机工作在低频时钟的情况下

当时钟频率低于 1.6GSPS 时, 建议在 SPI 启动过程中反转 CLKDIVPH 设置。由于延迟链的延时变化范围会超过一般的 4ns 窗口。假设最低的时钟速率为 800MSPS, 一个速率为 200MSPS 的 DCI 时钟对应 5ns 的时间, 超过了延迟链的长度。那么 4 分频电路 (DCO 输出) 的初始相位有可能会让数据接收控制器永远无法锁定。

如果时钟速率增加到 1600MSPS (DCI 时钟周期为 2.5ns), 则控制器总能找到至少两个 DCI 时钟边沿, 因而可以锁定。但是若要 DCI 的边沿对称的 (两边相等) 跟随初始的 DCI_DEL 中间刻度设置, 就需要 $\pm 0.75ns$ 的保护带。翻转 CLKDIVPH 可以让某一个 DCI 边沿向 DCI_DEL 中间刻度移动, 从而涵盖上此种

情况。

翻转 CLKDIVPH 可以让时钟以 TDAC/4 的补偿进行粗移。例如，在上述 800MSPS 和 1600MSPS 的情况下，将 CLKDIVPH 设为 1，分别对应了 5ns 和 2.5ns 的延迟。通过在数据接收控制器的 SPI 初始化过程中增加一个额外的步骤，可以增加延迟链的有效范围，从而保证 DCI_DEL 的值在一个合理的监控带内。

LVDS 驱动和接收电路

CBM97D39BG 集成了兼容 LVDS 的驱动器和接收电路。DCO 和 SYNC_OUT 的 LVDS 驱动器内部集成了等效的 200 欧姆源端电阻，在驱动外部 100 欧姆的负载是能保证输出电压摆幅位 +/-200mv 的信号电平。DCO 的输出可以通过寄存器 0x01 的第五位来控制使能。等效电路图如图 27 所示：

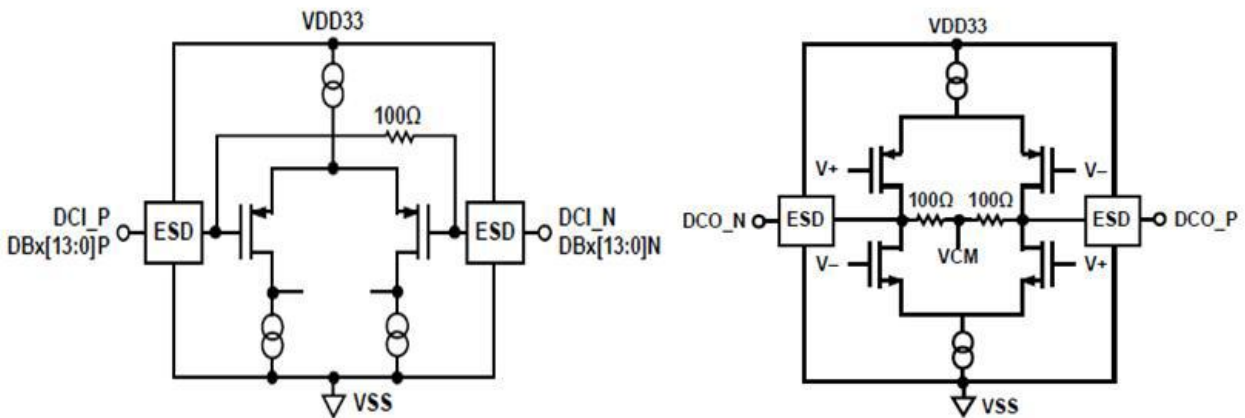


图 27 LVDS 输入和输出等效电路

如图 27 左边所示，CBM97D39BG 的 LVDS 接收器集成了 100 欧姆的负载电阻。接收器满足 IEEE-1596.3-1996 规范中有关输出摆幅的规格。LVDS 接收器可以通过寄存器 0x01 的 bit4 来控制使能。另外 CBM97D39BG 的 LVDS 输入端不具有失效保护能力，任何不需要的数据输入管脚都要通过外部网络连接固定的偏置或静态驱动器。如图 28 所示，对于没有用的 LVDS 输入管脚，通过电阻 R1 和 R2 将 Vp 和 Vn 分别设置为 1.4v 和 1v 的固定值。

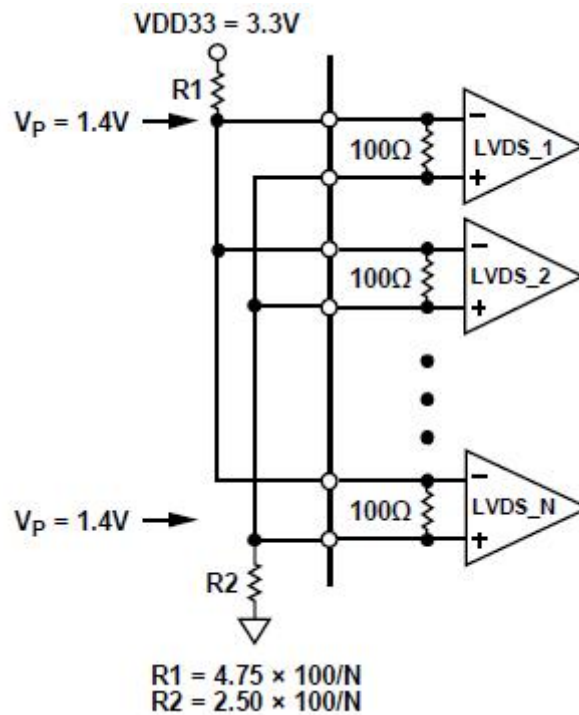


图 28 利用电阻网络将 LVDS 的输入设置为固定偏置

MU 控制器

CBM97D39BG 内部采用延迟锁相环 (DLL) 来优化器件内部数字域和模拟域之间的接口时序, 使数据在高达 2.5GSPS 的速率下能够成功传送到 DAC 的内核。如图 29 所示, DAC 时钟分为数字通道时钟和模拟通道时钟, 其中高要求的模拟时钟进入 DAC 内核 (抖动恶化最小), 而数字时钟进入一个可编程延迟链。延迟链的输出作为数字部分主时钟, 其它所有的数字域时钟, 包括内部数字时钟和输出的数字时钟, 均由其产生。延迟链的延迟量由 mu 控制器控制和优化两个时钟域之间的延时并连续跟踪其变化 (跟踪模式), 进而来保证数据在接口和器件内部各个模块之间能够正确传输。

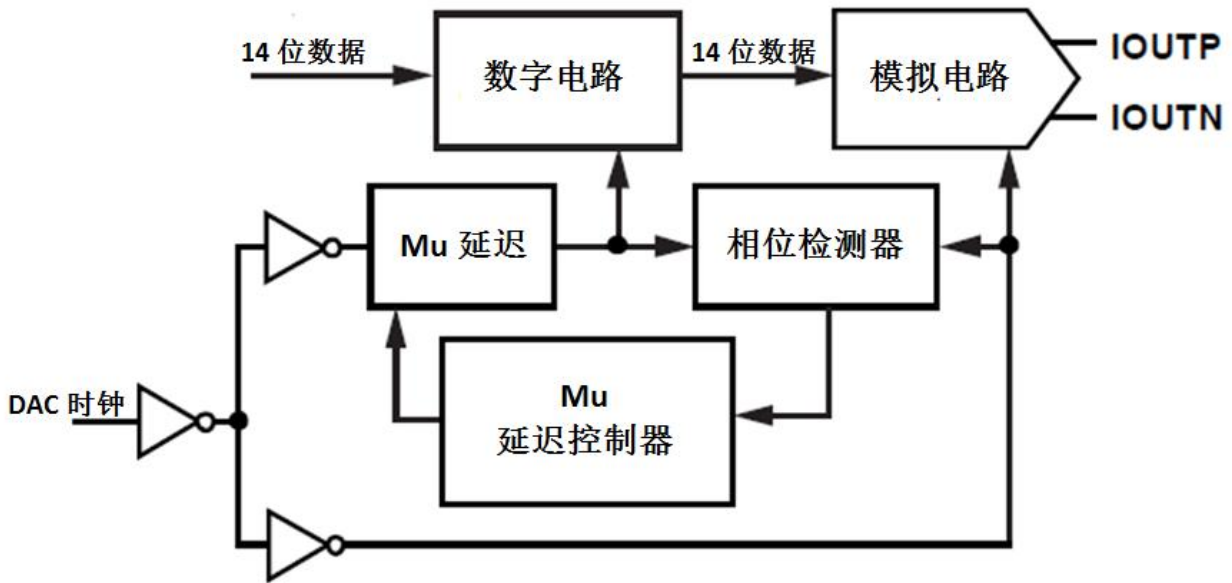


图 29 Mu 延时控制器的框图

Mu 控制器通过控制一个总延时为 864ps 的级联数字延迟链来调整数字域和模拟域时钟的时序关系。延迟值为 9 位分辨率, 可通过配置 MUDEL 寄存器来设置延时量, 范围为 0~432 (十进制), 精度为 2ps/LSB。由于时钟延迟可以对应到固定频率的时钟相位量, 控制环路通过比较两个时钟域的相位关系, 改变延迟线的延迟量来调整数字时钟的延时, 以满足模拟域时钟通路的固定相位延时需求 (SET_PHS)。

图 30 为 CBM97D39BG 工作在 2.4GSPS 时钟频率下 mu 典型相位特性与 9 位延迟控制字 (MUDEL) 的关系。Mu 相位范围值为 16, 对应 180 度。数字和模拟域之间的保留窗口为 0 (可以根据时钟速率扩展到 2)。选择合适的 mu 相位 (和斜率) 的目的是为了得到优越的交流输出特性, 并保证 mu 控制器对不同的器件都能锁定。例如, 如果工作在 1.6GSPS 到 2.5GSPS 之间时钟频率条件下, Mu 斜率和相位优化值为 -10, 那么在低于 1.6GSPS 时则需要设置为其它值。

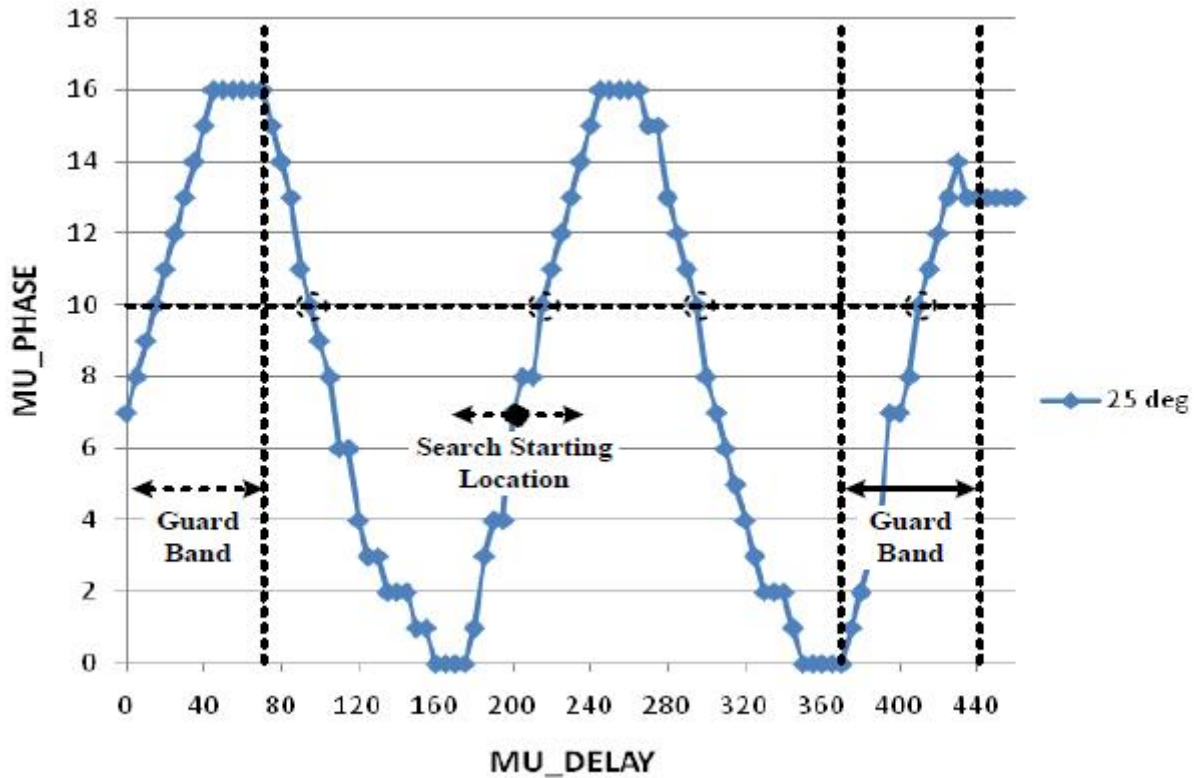


图 30 MU 相位特性与延迟的关系曲线

由于数字延迟链中器件跨导随工艺、温度、电源变化很大，不同器件之间的 Mu 相位特性偏差也较大。因此在使用过程中，需要小心选择目标相位的位置，使得 mu 控制器可以对不同的器件都能稳定在这个相位。图 31 为 800Mpsps 的工作时钟条件下，针对普通工艺条件及 25 度工作温度，低温、高温下，2 个器件的 mu 相位特性。注意到斜率相位值-8 在高温下对应不到任何延迟控制字，因此，在这个时钟频率下要使用另一个斜率相位值。

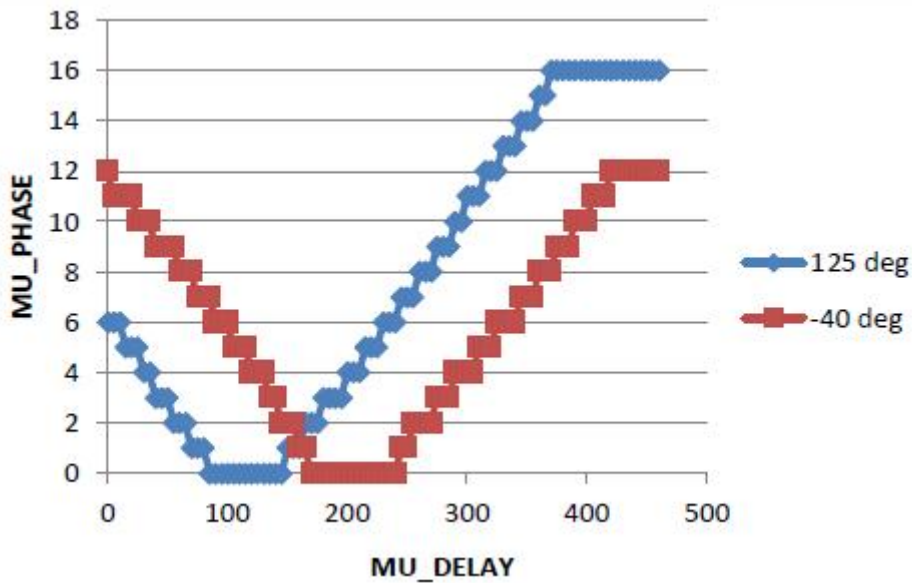


图 31 MU 的相位与延迟跟温度的变化曲线

基于上述考虑，表 26 给出了 CBM97D39BG 在特定时钟频率下的斜率/相位优化值。这些值能保证 mu 控制器很好工作。

表 26 Mu 相位值与时钟速率的关系

时钟速率 (GSPS)	斜率	MU相位
0.8	+	5
0.9	+	11
1.0	-	10
1.1	-	4
1.2	-	5
1.3	-	4
1.4	+	6
1.5	+	8
1.6 to 2.4	-	10

Mu 控制器完成搜索和锁定之后，能够在一定温度和电源电压范围内在两个时钟域之间保持一个固定的延时。如果 mu 控制器所需要的延迟超过了 Mu 延迟线的范围（也就是说， <0 或 >432 ）时，将会导致 Mu 控制器失锁，导致系统错误，从而产生中断信号 IRD 或者重启搜索。为了避免出现这种情况，每个 mu 延迟范围首尾边缘增加了两个对称的保护带。保护带范围设置寄存器 Guard[4:0]（寄存器 0x29）的

一个 LSB 对应 MUDEL (寄存器 0x28) 的 8 个 LSB。推荐的保护带设置为 11 (寄存器 0x29=0xCB) 对应 88 个 LSB, 从而提供足够的裕量。

MU 控制器初始化描述

在 SPI 启动过程中第一步, Mu 控制器必须完成初始并设置为跟踪模式。下面的步骤是 Mu 控制器的初始化过程:

- 1.启动时开启相位检测器 (寄存器 0x24=0x30)
- 2.开启 Mu 延迟控制器占空比控制电路, 定义预设斜率特性 (寄存器 0x25=0x80, 对应一个负的斜率)
- 3.定义搜索/跟踪模式、推荐相位(例如, SET_PHS 为 6)、MUDEL[8:0]起始值 216(寄存器 0x27=0x46、寄存器 0x28=0x6C)
- 4.设定搜索容差, 用于首次搜索失败后的重试。同样的, 设定监控带为 11 (寄存器 0x29=0xCB)
- 5.设置 Mu 控制器跟踪增益值, 并开启 Mu 控制器状态机 (寄存器 0x26=0x03)

最后一步完成后, Mu 控制器从 MUDEL 寄存器设定的初始值 (216, 对应延迟线的中点) 开始搜索。搜索算法通过交替扫描 Mu 延迟值, 直到找到所期望的相位 (SET_PHS 为 4)。当找到期望的相位值后, 通过计算得到被测相位的斜率, 并将该斜率与设定的斜率作比较 (斜率=负值)。

当一切匹配时, 搜索算法结束。如果不匹配, 搜索将在两个方向继续进行, 直到找到一个完全匹配点或者到达保护带。到达保护带后, 搜索将向反方向继续进行。如果反向搜索在达到保护带时仍然没有找到合适的相位点, 那么搜索模式将转换到交替模式并继续在保护带内搜寻。Mu 控制器典型的锁定时间大约为 180, 000 个 DAC 时钟周期 (在 2GSPS 工作时钟条件下大约为 75 μ s)。

如果 Mu 延迟控制器达到了结束点, 搜索就失败了。Mu 控制器可配置为重新搜索模式 (寄存器 0x29, 位 6) 或停止搜索模式。对于有微控制器的应用系统, 比较好的方法是在搜索时间耗尽后, 去查询 MU_LKD 的状态位 (寄存器 0x2A, 位 0), 检查是否锁定, 然后再尝试新的搜索 (通过写入 0x03 到寄存器 0x26)。另外, 这也利于系统控制器检查其余的系统参数状态 (比如电源和时钟源)。对于没有查询锁定状态能力的应用系统, 需要对 mu 控制器重新配置并重启搜索。

一旦找到 Mu 延迟值与 Mu 设置的理想相位和斜率精确匹配 (例如, 6 并且是负斜率), Mu 控制器便进入跟踪模式。在跟踪模式下, Mu 控制器会根据温度、时间以及供电电源的变化对延迟值作细微的调整来跟踪两个时钟路径的任何变化。内部寄存器 0x2A 的两个状态位, MU_LKD (寄存器 0x2A, 位 0) 和 MU_LST (寄存器 0x2A, 位 1) 提供给用户来标记已经存在的状态控制环路。如果目前的相位距离理想相位有四个以上的阶梯, MU_LKD 位会被清零。如果获得的锁定之前设置过, MU_LST 位就会被置位。相位偏移回到 3 个阶梯之内, MU_LKD 位会被置位, 同时 MU_LST 位清零。需要说明的是如果主要的时钟输入 (DACCLK) 被中断, 或者 mu 控制器超出了延迟链范围 (比如 <0 或 >432 的情况), 就会出现上述的情况。

如果出现失锁的情况，Mu 控制器可以选择保持在跟踪环路或者通过 CONTRST 位（寄存器 0x29，位 5）重新设定开始继续搜索。连续跟踪模式是一种很好的常用工作方式，因为对 CBM97D39BG 暂时失去锁定的系统来说是影响最小的。用户可以通过首先设置回读操作（寄存器 0x26，位 3），来访问 Mu 延迟和相位值。一旦设置为读操作，控制器正在使用的 MUDEL[8:0]位和 SET_PHS[4:0] 位（寄存器 0x27 和寄存器 0x28）可以经 SPI 口读出。

中断请求

CBM97D39BG 在内部一个或多个控制器锁定或失锁的情况下，可以向主处理器提供中断请求信号（IRQ），这些内部控制器包括 Mu 控制器、数据接收控制器和同步控制器。主处理器可以通过查询 IRQ 状态寄存器（0x04）的状态位来确定哪个控制器失锁。IRQ 输出为高电平有效信号，通过管脚 F13 输出到芯片外部。如果系统要使用到该中断请求信号，需要在外部经过一个 10k 欧的上拉电阻连接到 VDD33 上。

每一个中断请求 IRQ 可以通过设置 0x03 寄存器的相应使能位来使能该中断，同时在 0x04 寄存器有相应的中断标志位与其一一对应。这些中断状态位是在控制器满足中断条件的情况下被置位的。因此，对于 x_LCK_IRQ 和 x_LST_IRQ 状态位可能会在控制器暂时失锁的情况下置位，但是主处理器还没来得及处理该中断请求的情况下又重新锁定。在这种情况下，主处理器需要通过回读 0x21 或 0x2A 的当前状态位来验证当前的状态，然后主处理器通过判断当前的状态来采取相应的操作，比如重新建立锁定等。在完成中断处理后，需要将 0x03 的中断标志位清除，具体的操作方法是向 0x03 的相应位先写 0，然后再写 1 就可以清除掉中断标志位。图 32 提供了详细中断电路的原理框图：

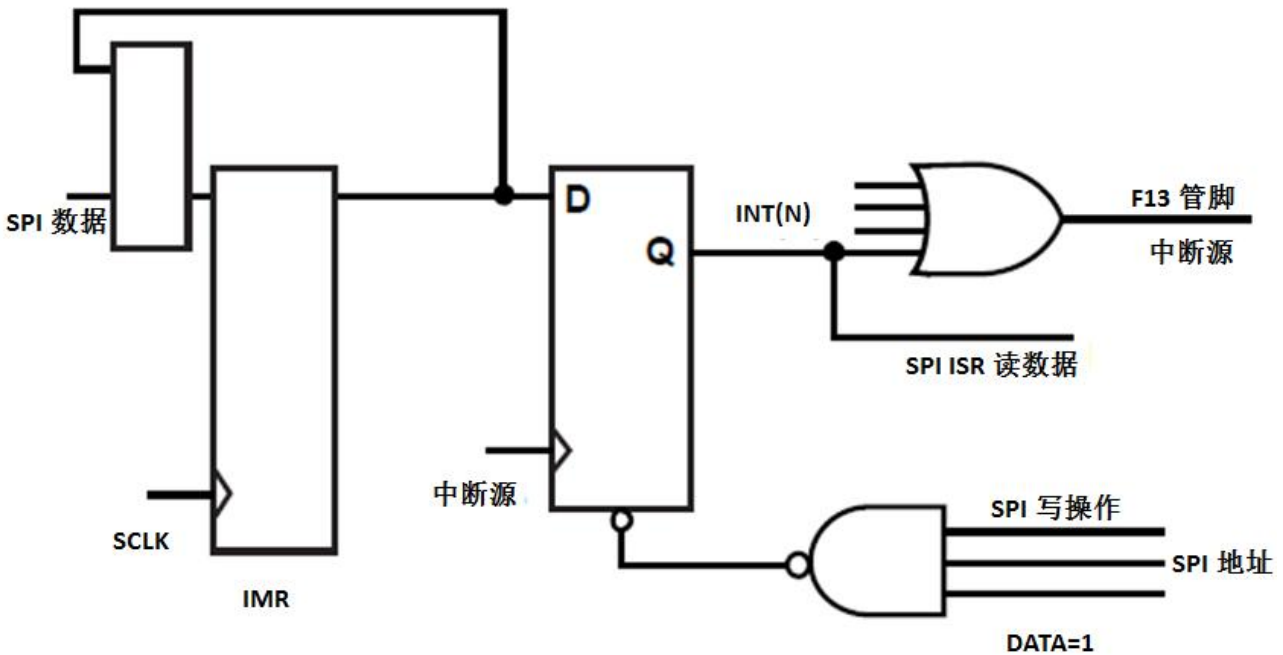


图 32 中断请求电路图

另外,也可以在 CBM97D39BG 上电初始化完成后,利用中断请求来判断 Mu 控制器和数据接收器是否锁定。例如,在使能 Mu 控制器之前,将 MU_LCK_EN 位 (0x03 寄存器的 bit2) 置位,这样的话,IRQ 中断输出信号会监控锁定状态,并决定是否进一步对数据接收控制器进行相应的操作。需要注意的是在对下一个控制器进行操作前,需要将相关的 LCK 位清零。等到所有的控制器都锁定后,才能将相应的失锁使能位 (x_LST_EN) 置 1。有关中断请求寄存器的详细描述,请参考表 27。

表 27 中断请求寄存器功能描述

寄存器地址 (十六进制Hex)	位 (bit)	说明
0x03	5	SYNC_LST_EN
	4	SYNC_LCK_EN
	3	MU_LST_EN
	2	MU_LCK_EN
	1	RCV_LST_EN
	0	RCV_LCK_EN
0x04	5	SYNC_LST_IRQ
	4	SYNC_LCK_IRQ
	3	MU_LST_IRQ
	2	MU_LCK_IRQ
	1	RCV_LST_IRQ
	0	RCV_LCK_IRQ
0x21	7	SYNC_TRK_ON
	5	SYNC_LST
	4	SYNC_LCK
	3	RCVR_TRK_ON
	1	RCVR_LST
	0	RCVR_LCK

0x2A	1	MU_LST
	0	MU_LKD

多芯片同步

多个 CBM97D39BG 的同步需要所有的器件有相互匹配的流水线延迟。这意味着所有的器件在相同的时间点上，在相同的输入数据条件下，各个 DAC 的输出信号在时域上相位相互对齐。不同器件之间相位模糊主要来源于驱动 Rx 接收数据路径和数据控制器（见图 34）的 4 分频电路。这个相位的不确定性会引起任意两个器件之间 ± 2 个样点偏移。由于内部分频器的初始状态在上电时是未知的，就需要有多个 CBM97D39BG 器件的数字通路同步手段来确保器件内部流水线延迟一致。

图 33 为多个 CBM97D39BG 同步顶层框图，假设 FPGA（或在多个 FPGA 中）内部不同数据流在送到不同 DAC 之前已经很好的同步。一个共用的 RF 时钟源经过一个时钟缓冲器（例如 ADCLK946）后经由匹配的 PCB（等长的 PCB 布线及匹配电路）线路传输到不同 CBM97D39BG 的时钟输入口。其中一片 CBM97D39BG 工作在主模式并输出 SYNC_OUT 同步参考时钟（频率为 $f_{DAC}/4$ 给自身和其它工作在从模式的 CBM97D39BG 的 SYNC_IN 使用。主模式的 CBM97D39BG 的 SYNC_OUT 和 DCO 输出经过不同的 LVDS 缓冲器后通过匹配的延迟线路分别输入到所有 DAC 的 SYNC_IN 口和 FPGA。以确保各片 DAC 和 FPGA 严格同步。如果只有一片 FPGA 的情况下，主模式 CBM97D39BG 的 DCO 可以直接用于驱动 FPGA。

同步完成后，不同器件内部的 4 分频器将保持相同的输出信号相位去驱动各自的 LVDS 控制器。所有器件的 Mu 控制器和数据接收控制器必须配置成相同的 SPI 值，也就是说 SET_PHS 和 DCI_DEL 的配置必须一致。这样可以确保不同器件的控制器具有相同的延时。为了验证各个器件内部延迟相同，可以通过回读所有器件的延时值（MUDEL 和 DCI_DEL）来验证。

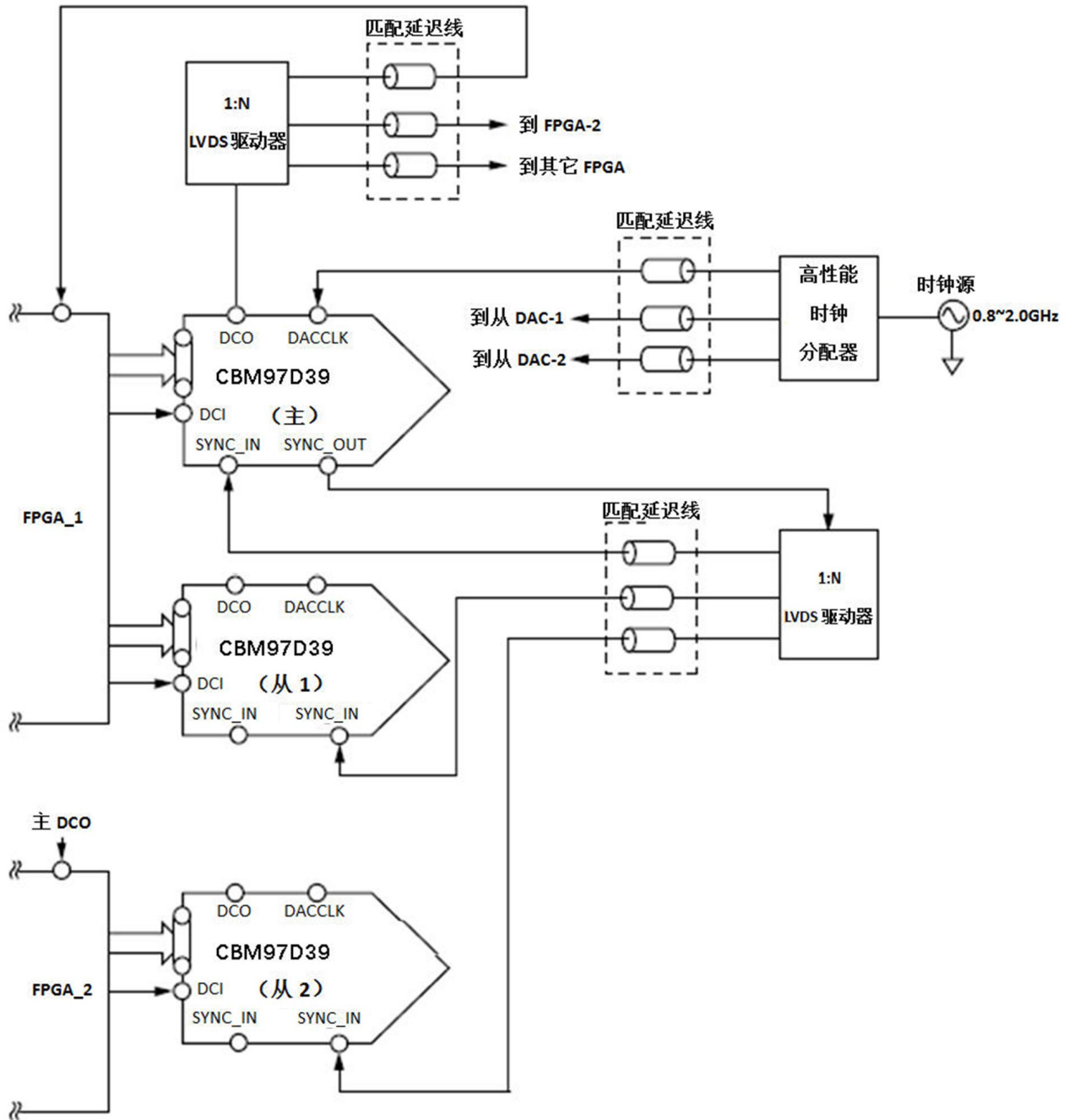


图 33 多器件同步方案框图

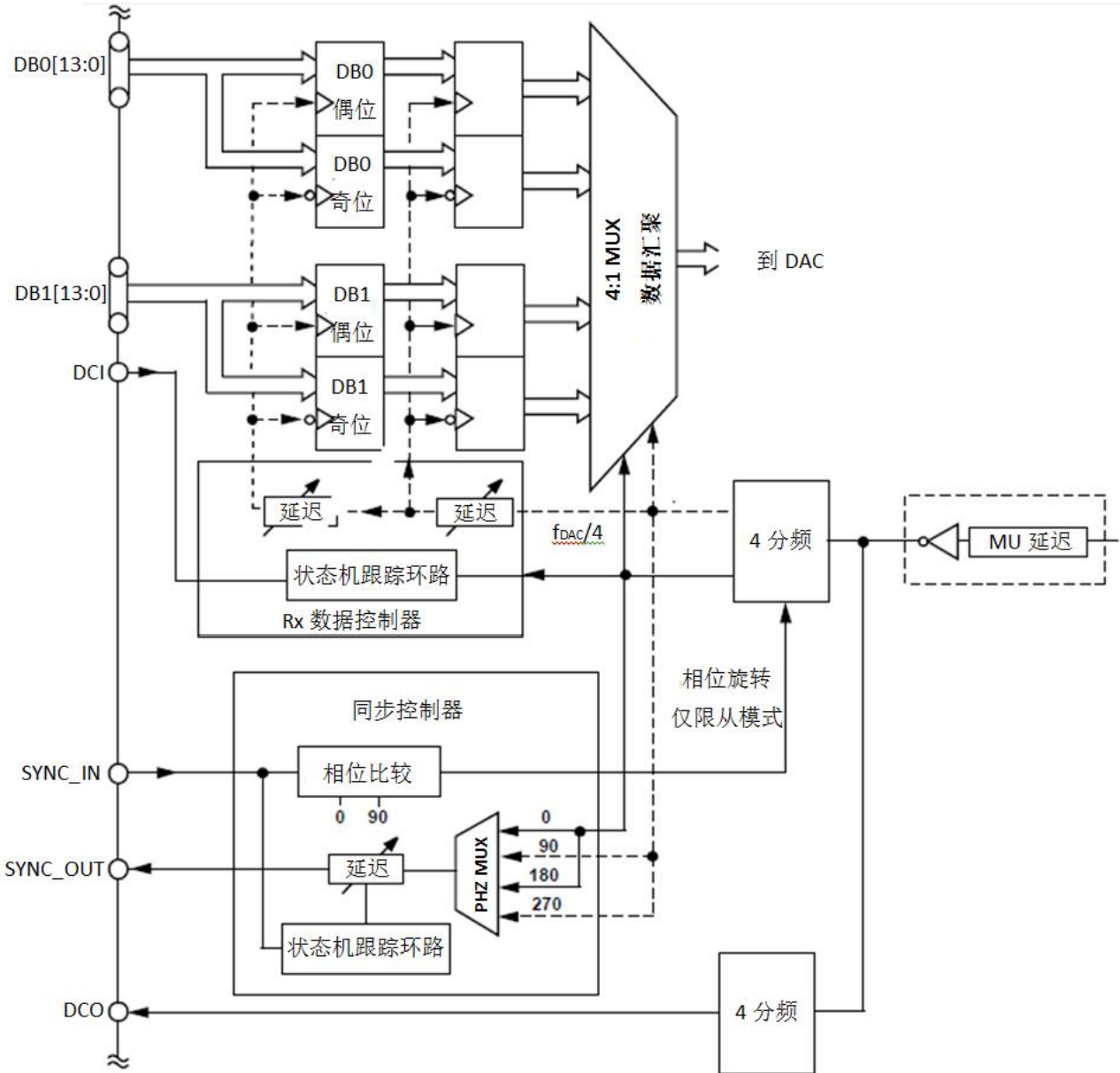


图 34 同步电路和控制器框图

图 34 的底端为一个同步控制器的框图，从该部分框图可以看出同步控制器是如何跟其他数字功能模块进行接口的。从上面功能框图可以得到以下信息：

- 多个器件之间的同步可以通过调整从器件内 4 分频器的相位来实现与主器件对齐。
- 对于从器件，同步控制器比较主级 SYNC_IN 参考信号的相位对齐和除 4 分频器的初始 0°到 90°输出。然后调整 4 分频器的相位，直到 SYNC_IN 信号在这些相位之间。
- 需要有一个公共的参考信号才能实现多器件的同步。主器件产生并输出 SYNC_OUT 信号，提供给自身和其它从器件并作为 SYNC_IN 信号分配到所有器件（包括自身严格的延时对齐）。

- 由于 SYNC_IN 信号与主器件的除 4 分频器之间有确定的相位关系，所有的从器件内的除 4 分频器的相位可以分别跟 SYNC_IN 相位对齐，从而确保所有的器件相位对齐。
- 由于无法通过使能同步控制器来手动调整数据路径上的除 4 分频器的相位。这在低时钟速率下会成为一个问题，当期望通过调整除 4 分频器的相位来保证数据接收控制器的锁定和/或达到一个更优化的 DCI_DEL 值。
- DCO 输出信号是从另外一个除 4 分频电路产生，因此每次启动时输出信号的相位是一个随机相位。由于这个原因，主级的 DCO 应该被分配到所有的 FPGA。

同步控制器的初始化说明

主芯片同步控制器通过写入 0x70 到寄存器 0x10 来使能的。使能后，内部会自动调整 SYNC_OUT 信号的输出延时，使反馈信号 SYNC_IN 信号落在除 4 分频器输出信号的 0°和 90°的正中间。延时粗调是通过 PHZ MUX 相位切换实现，延时细调是通过可变延时链来实现。可变延时链的调整步长为 12ps。一旦 SYNC_IN 处于分频器 0°和 90°输出的正中间，控制器就进入跟踪模式，使得 SYNC_IN 在电源和环境温度变化时，始终能维持在分频器相邻相位时钟输出的正中间。因此可以保证各从芯片的 SYNC_IN 也处于其各自除 4 分频器 0°和 90°输出的正中间，从而提供了最大的采样裕度来容忍非理想延时偏差。锁定、失锁和跟踪状态可通过寄存器 0x21 的状态位来读取：SYNC_LCK, SYNC_LST, SYNC_TRK_ON。

从芯片同步控制器通过写入 0x50 到寄存器 0x10 的方式来使能。使能后，芯片内部的状态机会对参考同步输入信号 SYNC_IN 的相位和除 4 分频器的 0°/90°输出相位进行比较。如果 SYNC_IN 信号相位没有落在这两个相位之间，控制器就会调整除 4 分频器的输出相位，直到 SYNC_IN 的相位位于 0°/90°之间为止。为了验证相位对准与否，可以通过回读寄存器 0x0D 的 bit[5:4] (SYNC_IN_PH90 和 SYNC_IN_PH0) 来确认，如果寄存器 0x0D 位[5:4]，分别为 1 和 0，则表示 SYNC_IN 位于 0°/90°之间。另外从器件的 DCO 和 SYNC_OUT 输出信号可以通过寄存器 0x01 的 bit5 的设置屏蔽掉。

同步限制

在更高的速率下为保证不同批次的两片或多片 CBM97D39BG 之间的同步难度很大，因为除 4 分频器的输出时钟的相邻相位之间延迟失调等于 $1/f_{DAC}$ 。例如，对 2GSPS 工作时钟的 DAC，一个周期对应的时长为 500ps。在这种工作条件下，如果理想主级器件的 SYNC_IN 信号位置在 div-by-40°和 90°相位输出的中心的话，只留下 $\pm 250ps$ 的定时裕量给从器件。这个理想的裕量受主级除 4 分频电路中正交相位误差的影响和器件内部电路将 SYNC_IN 准确定位在 0°和 90°的输出相位之间的能力等两方面的影响而减少。

下列因素会进一步恶化定时裕量：

- 主-从器件在 Mu 延迟时钟路径和 SYNC_IN 增加延时的失配。这些器件间在极端工艺下的失配最大可以到 100ps；
- 从级除 4 分频器输出信号的正交相位误差。

这些影响时钟延时偏差的因素随着 DACCLK 的周期缩短而变得更加严重，从而减少了主-从器件的外部时钟偏差的裕量。这样的话，用户在 PCB 的设计和时钟分配器的选择上需要格外注意，确保 DACCLK 和 SYNC_IN 信号之间的偏差最小。另外需要考虑到对于 FR4 的 PCB 上，线路延迟增量大约为 170ps/英寸，以及每个时钟分配 IC 的不同输出之间偏差可能高达 25ps。

对于多个板级之间的同步，由于 DACCLK、SYNC_OUT 和 DCO 需要经过背板分配到不同的 PCB 上，同步问题变得更具有挑战性。除了要保证主器件输出的 DCO 经过背板在不同板级之间需要同步外，还需要对不同的数据源进行同步。而这些数据源（如不同的 FPGAs）来自于不同的 FPGA，由于工艺、供电电压和温度敏感性不一致（PVTs），可能导致各自 DCI 输出信号的失配。

即使是 DCI 输入严格对齐，由于延迟链上 PVT 影响，不同 CBM97D39BG 内部的数据接收控制器锁定在不同的延迟控制字。如果绝对的延迟超过了 $4/f_{DAC}$ 周期，就可能导致不同器件之间存在 4 个样点的流水线失配。由于控制器从初始值（DCI_DEL 和 SMP_DEL）开始上下搜索第一个有效边沿。如果器件之间的初始值相同，但是在不同的 PVT 下由于绝对延迟的不同，可能使芯片锁定在初始值上下的不同 DCI 边沿。因此，必须确定多芯片之间的 DCI_DEL 值匹配，使 DCI_DEL 值的绝对差不超过 1 个数据周期（ $4/f_{DAC}$ ）。如果差值超过了 1 个数据周期，修正从器件的 DCI_DEL（SMP_DEL）设置，使得它的初始点大约为主级和从级读回值差的 1/2。

模拟接口

模拟工作模式

如图 35 所示，CBM97D39BG 采用四开关架构。在半时钟周期，仅有一对开关使能，因此需要在交替的时钟沿上使用各对开关。四开关架构的主要优势是可消除传统双开关架构 DAC 输出信号中出现的与码型相关的毛刺。

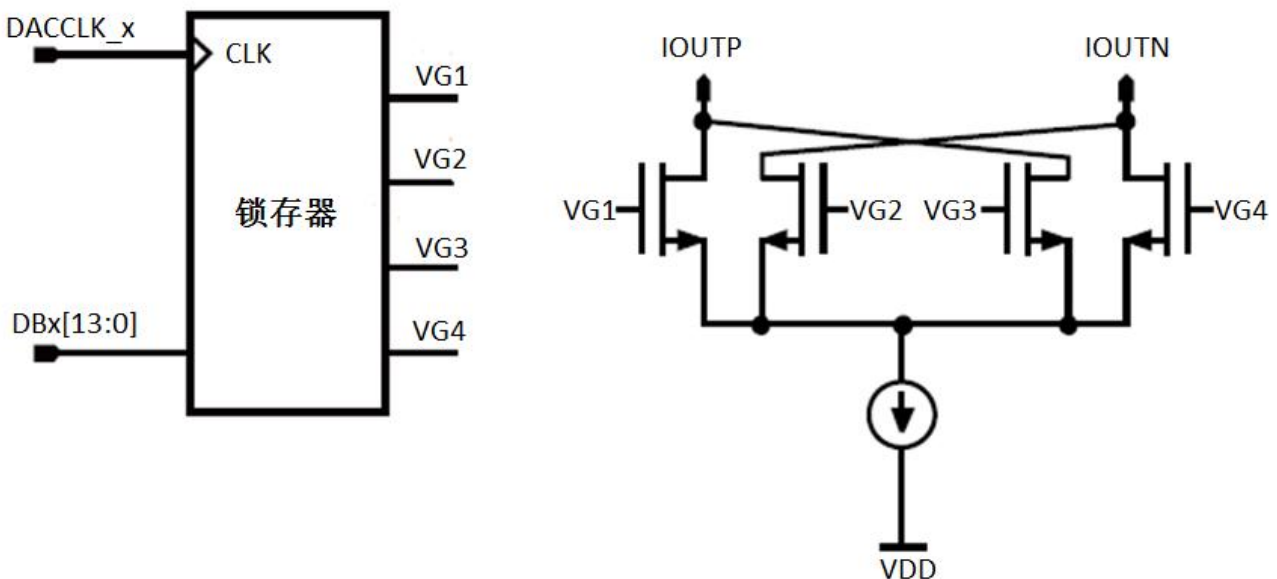


图 35 CBM97D39BG 的四开关模拟输出架构

在双开关架构中，当开关转换发生且 D1 和 D2 处于不同状态时，就会产生毛刺。然而，如果 D1 和 D2 处于相同的状态，则开关转换不会引起毛刺。这种码相关毛刺会增加 DAC 的失真量。在四开关架构中，无论输入是什么样的码，总有两个开关在半时钟周期处转换，因此不会产生码相关毛刺，而是产生较大的 $2 \cdot f_{DACCLK}$ 时钟杂散。

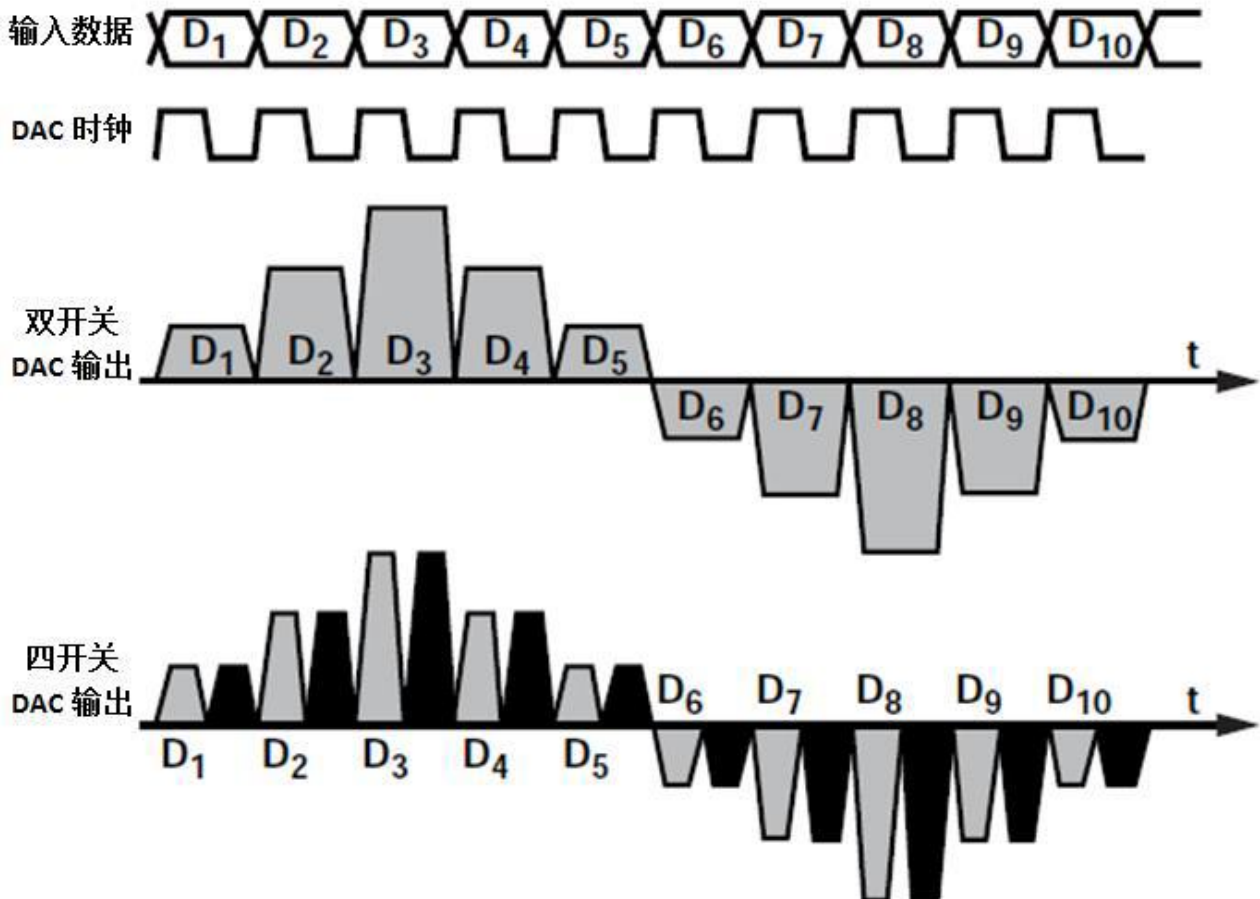


图 36 双开关和四开关 DAC 输出波形

4 开关 DAC 的另外一个特性是 DAC 的内核可以工作在三种模式：正常模式、混频模式（Mix 模式）和归零模式（RZ 模式）。可以通过 SPI 寄存器 0x08 的 bit[1:0]来设置 CBM97D39BG 的工作模式，上电复位后，默认为正常模式。使用混频模式时，器件以 DAC 时钟速率对输出信号进行有效削波。其作用是降低基频信号的功率，从而提高镜像信号的输出功率。归零（RZ）模式与混频模式基本类似，只不过是中间样点被 DAC 输出的中点值代替。有关混频模式和归零模式的输出波形，参考图 37 所示。

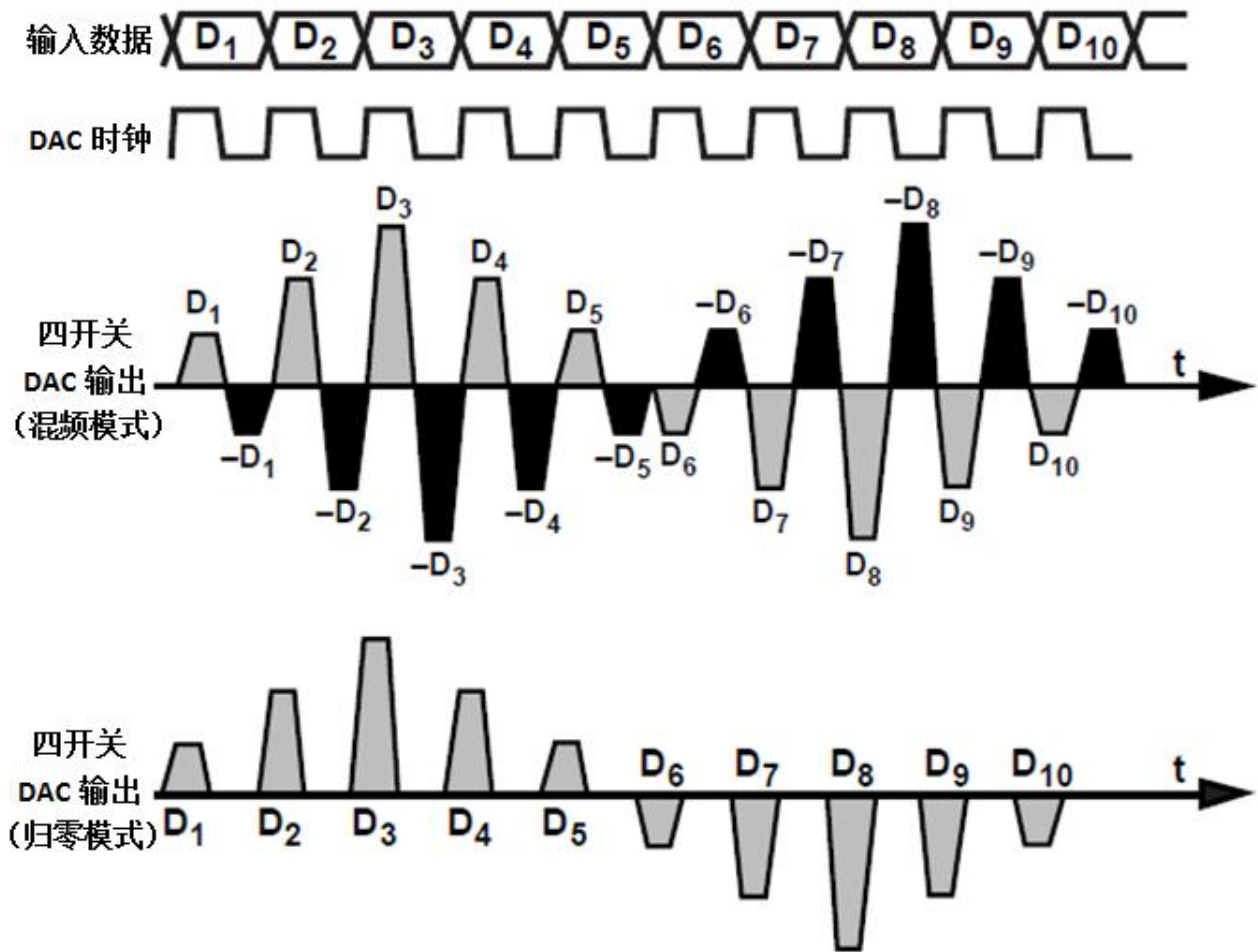


图 37 混频模式和归零模式下 DAC 的输出波形

相对于混频模式, 归零模式的输出信号功率有 6dB 的损失, 这是因为 DAC 只有 1/2 的时钟周期工作。CBM97D39BG 提供这几种工作模式, 非常方便用户在第一到第三个奈奎斯特区内灵活产生所需要的载波信号。在不同模式之间切换的时候, DAC 的输出 SINC 函数滚降特性也会随之发生改变, 在不同的奈奎斯特区域内, 能得到的最大输出信号功率取决于载波的位置。在实际应用中, 特别是在 DAC 工作时钟很高的情况下 ($f_{DAC} > 2G\text{sps}$), 第三奈奎斯特区域内可用的信号带宽是比较有限的。这是由于受到 DAC 内核输出带宽的影响, 以及外部接口网络的影响, 比如外部 Balun 带来的额外滚降的影响。图 38 显示了在不同工作模式下的 DAC SINC 函数的滚降特性。

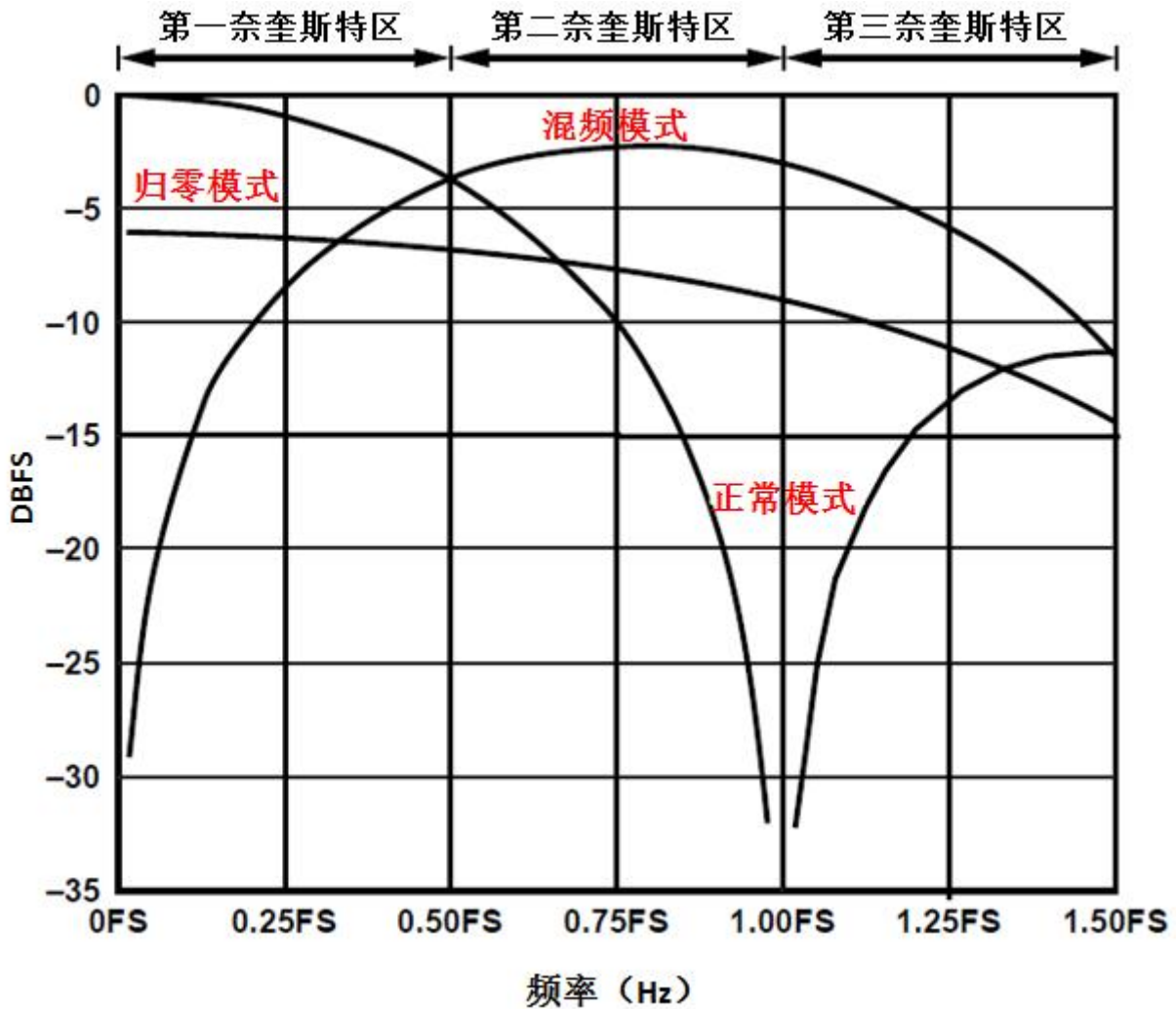


图 38 CBM97D39BG 在不同模式下 SINC 函数的滚降特性

时钟输入

时钟信号的质量以及合适的电平会直接影响到 DAC 的交流输出特性。所选用的时钟源的相位噪声和杂散特性应该满足目标应用的要求。时钟源指定频率偏移处的相位噪声和杂散会被直接转换到输出信号中。可以证明，当 DAC 时钟路径对时钟性能影响可以忽略不计时，经 DAC 重建输出的正弦波的相位噪声特性与时钟源的关系是 $20 \times \log_{10}(f_{OUT}/f_{CLK})$ 。CBM97D39BG 内部集成了高性能的时钟接收器，在外部输入时钟功率低至 0dBm 的时候，仍然能得到很好的 DAC 输出性能。图 39 给出了一个典型的 DAC 时钟输入参考电路，该电路保证在时钟输入低至 0dBm 时，仍能保证较好的 DAC 性能。

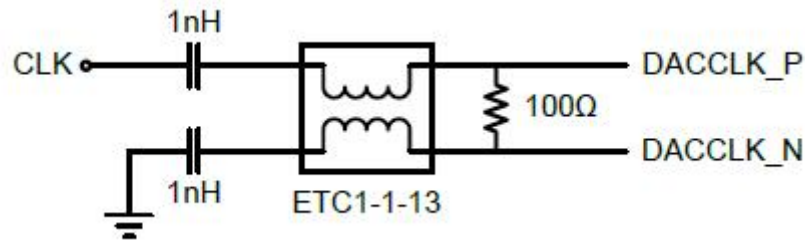


图 39 CBM97D39BG 的典型时钟输入参考电路

电压基准

CBM97D39BG 的输出电流能用一组数字控制位和 I120 基准电流来控制，如图 40 所示。将带隙电压施加于 I120（管脚 B14）与地之间的 10K 欧外部电阻上便获得是由带隙基准电压外接一个 10 kΩ 的电阻产生到地的 I120（管脚 B14）获得 120uA 的基准电流。

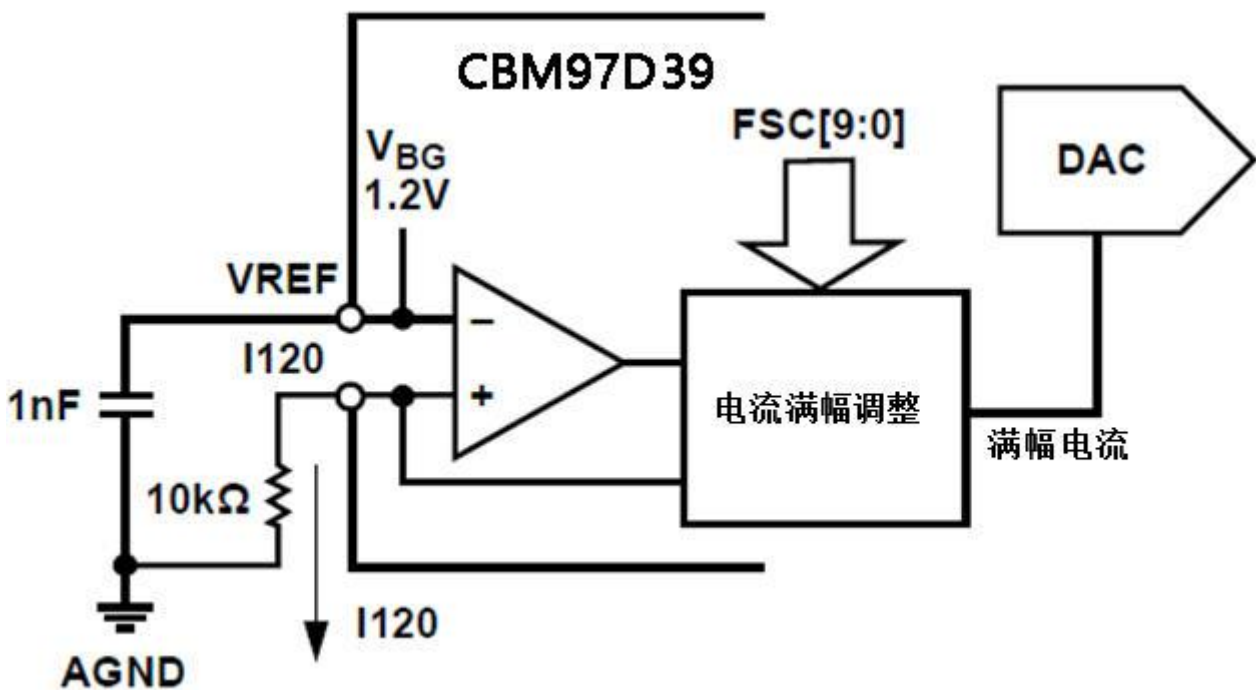


图 40 基准电压源电路

配置基准电压源要注意以下几点：

- 为了使器件正常工作，10k 欧的电阻和 1nF 的旁路电容均是必须的；
- 将 DAC 输出满量程电路 IOUTFS 调整为默认设置 20mA 以外的值时，需要通过 SPI 寄存器，用数字调节的方式实现；
- CBM97D39 不是乘法 DAC，不支持用交流信号对 120uA 的基准电流进行调制；
- VREF 引脚上的带隙电压必须经过缓冲才能用于外部电路，输出阻抗约为 5K 欧；

- 可以将一个外部基准源连接到 VREF 引脚上。

如上所述, IOUTFS 可以通过寄存器 0x20 的[7:0]和 0x21 的[1:0] (FSC[9:0]) 位, 在 8.7mA 到 31.7mA 范围内进行数字调整。下式反映了 IOUTFS 与 FSC[9:0] (设置范围是 0 到 1023) 之间的映射关系:

$$I_{OUTFS} = 22.6 \times FSC[9:0]/1000 + 8.7$$

注意, 默认值 0x200 产生 20mA 的满量程电流, 除非特别说明, 本手册中有关 DAC 输出的特性参数均是基于该配置得到的结果。

模拟输出

DAC 的等效输出电路和传递函数

CBM97D39BG 提供互补的电流输出 IOUTP 和 IOUTN, 内部的电流经过参考负载到地。图 41 显示了 DAC 的等效输出电路。与这种类型的大多数电流输出 DAC 相比, CBM97D39BG 的输出存在微小的偏移电流 (大小为 $I_{OUTFS}/16$), 峰值差分交流电流略低于 (大小为 $15/32 \times I_{OUTFS}$)。

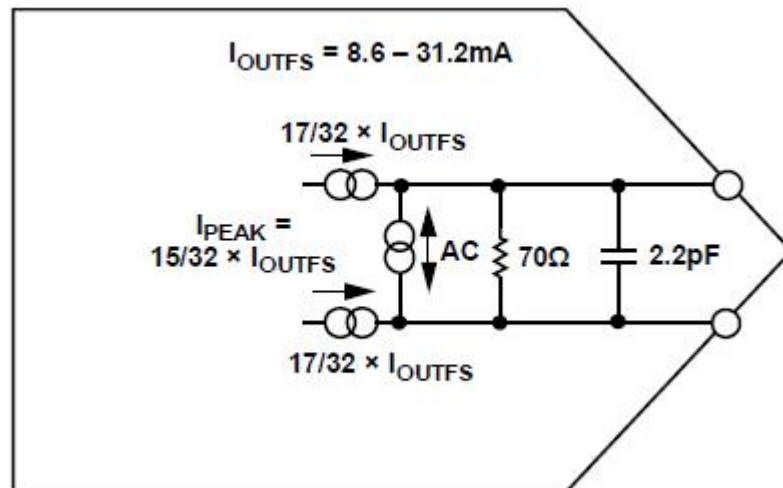


图 41 CBM97D39BG 的输出等效电路

从图 41 可以将 DAC 的输出看着是一对直流电流源模型, 它们各自提供输出 $17/32 \times I_{OUTFS}$ 的电流。一个差分的交流电流源 IPEAK 用于模拟 DAC 的输出信号。此交流电流源的极性和信号的相关性与数字码 (F) 的关系如下所示:

$$F(\text{CODE}) = (\text{DACCODE} - 8092)/8092$$

$$-1 \leq F(\text{CODE}) < 1$$

其中, DACCODE = 0 到 16384 (十进制)。

由于 IPEAK 的摆幅为 $\pm (15/32) \times I_{OUTFS}$, 因此 IOUTP 和 IOUTN 上测得的输出电流可在 $I_{OUTFS}/16$ 到 I_{OUTFS} 范围内变化。然而, 由于交流信号相关电流成分互补, 因此两个输出之和始终未常数, 即 $I_{OUTP} + I_{OUTN} = (34/32) \times I_{OUTFS}$ 。IOUTP 和 IOUTN 输出端得到的码相关电流表示为:

$$IOUTP = 17/32 \times IOUTFS + 15/32 \times IOUTFS \times F(\text{CODE})$$

$$IOUTN = 17/32 \times IOUTFS - 15/32 \times IOUTFS \times F(\text{CODE})$$

图 42 显示了 IOUTFS 设置为 19.65mA 时 IOUTP 与 DACCODE 之间的传递函数。

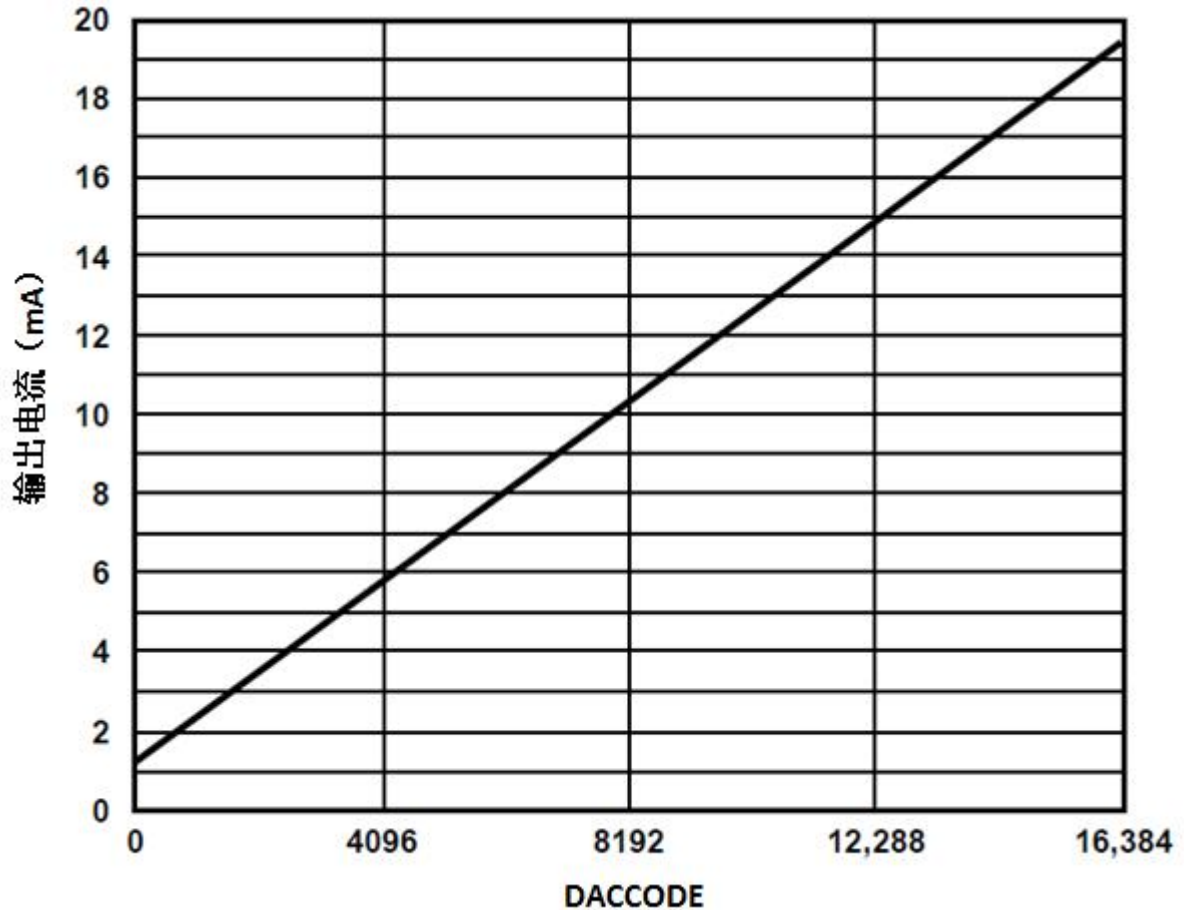


图 42 增益曲线: FSC[9:0] = 512, DAC 偏移= 1.228mA

DAC 的峰值输出功率

差分电流输出 DAC 的最大峰值功率能力取决于峰值差分交流电流 IPEAK 和等效负载电阻。由于 CBM97D39BG 内部等效为 70 欧姆的差分阻抗，在输出需要连接具有源端阻抗为 50 欧姆的 1:1 巴伦的情况下，需要匹配到 50 欧姆的等效输出，如图 43 所示。

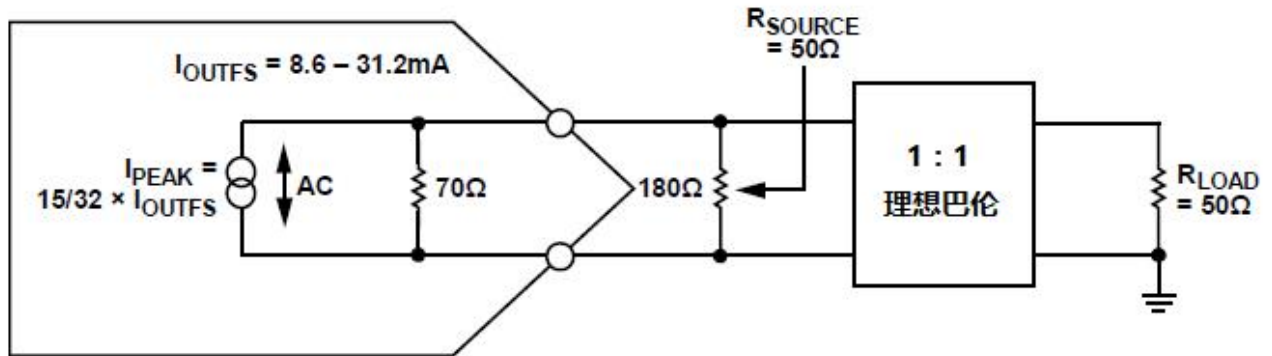


图 43 在外部 50 欧姆负载得到最大功率的等效电路图

如果 CBM97D39BG 设置为 $I_{OUTFS} = 20\text{mA}$ 输出，其峰值交流电流为 9.375mA ，提供给等效负载的峰值功率为 2.2mW ，即 $P = I^2R$ 。由于 1:1 巴伦的源电阻和负载电阻相等，因次功率由二者均分，输出负载获得 1.1mW 或 0.4dBm 的峰值功率。

要计算供给负载的均方根功率，需要考虑一下事项：

- 数字波形的峰值与均方根之比；
- 相对于满量程的数字信号回退；
- DAC 的 SINC 函数响应和外部网络中的非理想损耗。

例如，一个没有数字回退的重构正弦波在理想情况下应该测的 -2.6dBm 的功率，因为其峰均比为 3dB 。如果包括典型的 0.4dB 的巴伦损耗，不考虑 DAC SINC 函数衰落影响，用户可得到预期 -3dBm 的实际功率。要想得到更大的输出功率，最好的办法是通过提高 I_{OUTFS} 来实现。

输出级配置

CBM97D39BG 主要针对需要支持宽信号带宽（比如基于 DOCSIS 的 CMTS 系统或宽带雷达系统）和/或高 IF/RF 信号合成的高动态范围系统而设计的。只有在下列条件下，才能得到最佳的交流输出特性：DAC 输出配置为差分（即平衡）工作模式，且输出共模电压偏置到模拟地。DAC 的输出接口网络需提供接近 0 欧姆的直流偏置通路连接到模拟地上。在整个输出频率范围内， I_{OUTP} 和 I_{OUTN} 引脚之间的任何输出阻抗不平衡都会降低失真性能（主要是偶次谐波失真）和噪声性能。另外，外部元器件的选择和 PCB 的布局布线也会起到至关重要的作用。图 44 为一典型的宽带应用示意图，能覆盖最高达 2.2GHz 的输出应用。

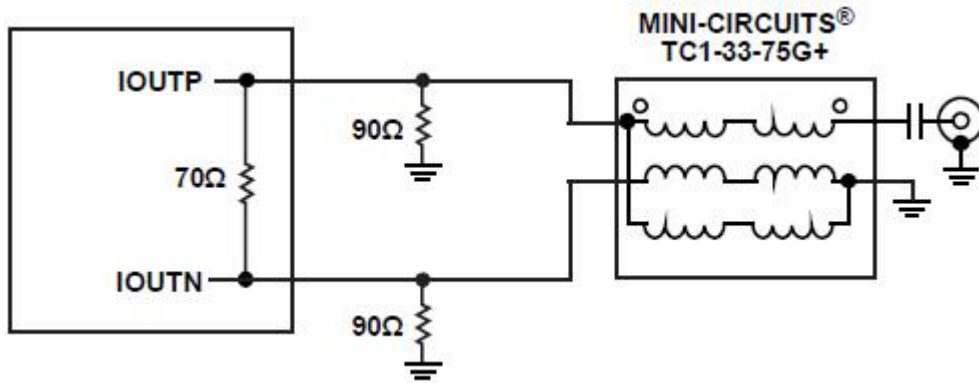


图 44 高达 2.2GHz 输出的推荐电路

图 45 给出了可以直接跟 DAC 的输出连接，带自偏置的差分增益模块接口方式。电感 (L) 用着 RF 扼流圈，提供直流偏置路径以连接到模拟地 (AGND)。电感与隔直电容共同决定复合通带响应的截至频率下限。由于许多差分放大器是由两个增益匹配的单端放大器组成，因而无法提供共模抑制，而且由于匹配不佳，还可能降低平衡性，为了得到更好的系统性能，在差分放大器的前端可以加上一个 RF 巴伦，它能够有效抑制从 DAC 输出端来的共模噪声、谐波以及时钟杂散等进入到放大器。

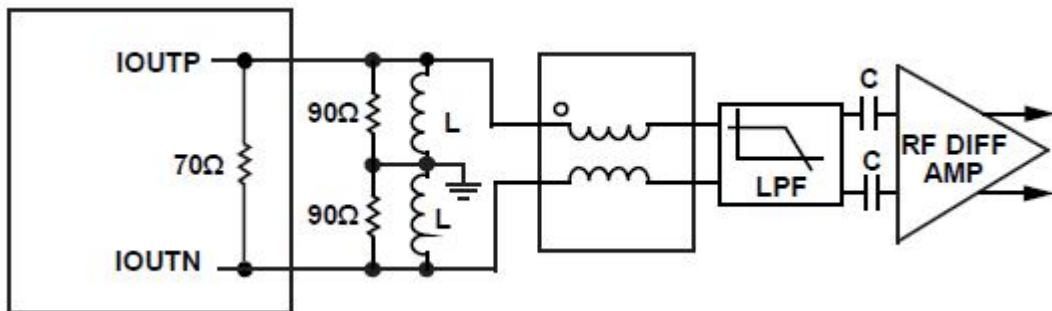


图 45 DAC 的输出与自偏置的差分增益模块接口电路

对于混频模式下，要想 CBM97D39BG 输出频率超过 2.2GHz，用户可以考虑图 46 所示的电路。该电

路采用宽带巴伦自偏置的配置方式。

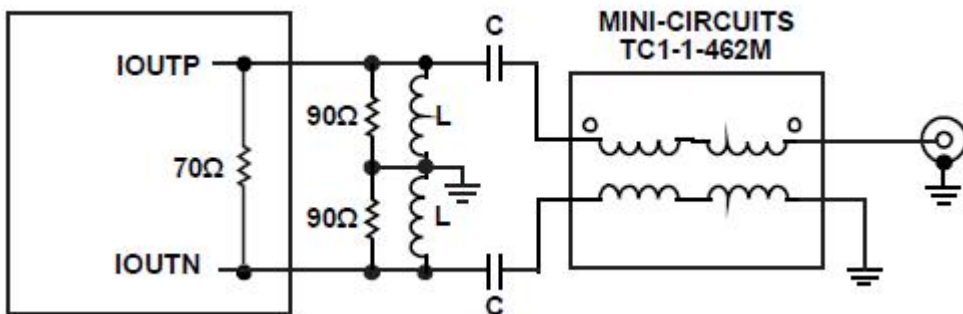


图 46 在混频模式下推荐的接口电路

启动过程推荐

CBM97D39BG 上电后，主处理器需要通过 SPI 口对其进行初始化配置。图 47 为初始化的流程图，表 28 和表 29 详细列出了流程图中每一步操作需要的 SPI 寄存器的读/写操作，说明如下：

- 软件复位不是必须的步骤，因为 CBM97D39BG 内部集成了上电复位电路和提供外部硬件复位管脚 RESET；
- SYNC 控制器是可选配置过程，因为该功能仅仅实用与两片或更多片器件之间的同步。如果需要同步，仍然可以通过比较不同器件的 DCI_DEL 的值是否匹配来验证；
- Mu 控制器必须首先使能且工作在跟踪模式下，然后才能使能数据接收控制器，因为 DCO 输出信号来自该部分电路；
- 等待周期是相对于 fDATD 的周期而言的；
- 限定尝试锁定控制器的次数不超过 3 次，典型情况下，一次尝试就可以完成锁定；
- 可以使用硬件或软件中断的方式来监测控制器的状态。

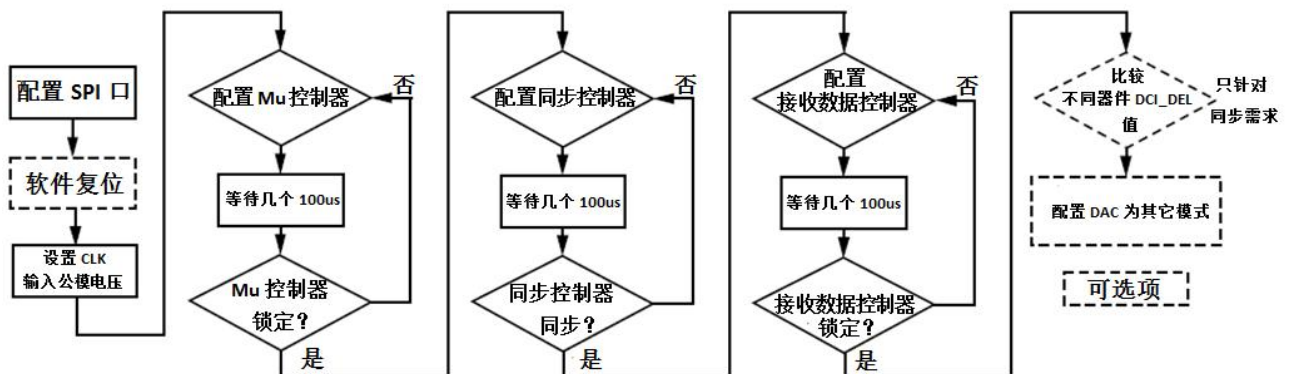


图 47 CBM97D39BG 的初始化配置流程

表 28 推荐的 SPI 初始化流程 (SYNC 关闭)

步骤	寄存器地址 (Hex)	寄存器值	注释
1	0x00	0x00	配置四线 SPI 模式，位[7:5]必须被镜像到[2:0]，因为在上电时 MSB/LSB 格式未知。
2	0x00	0x20	软件复位到 SPI 的默认值
3	0x00	0x00	清除复位
4	0x22	0x0F	设置输入差分时钟 DACCLK_P 和 DACCLK_N 的共模电压。
5	0x23	0x0F	

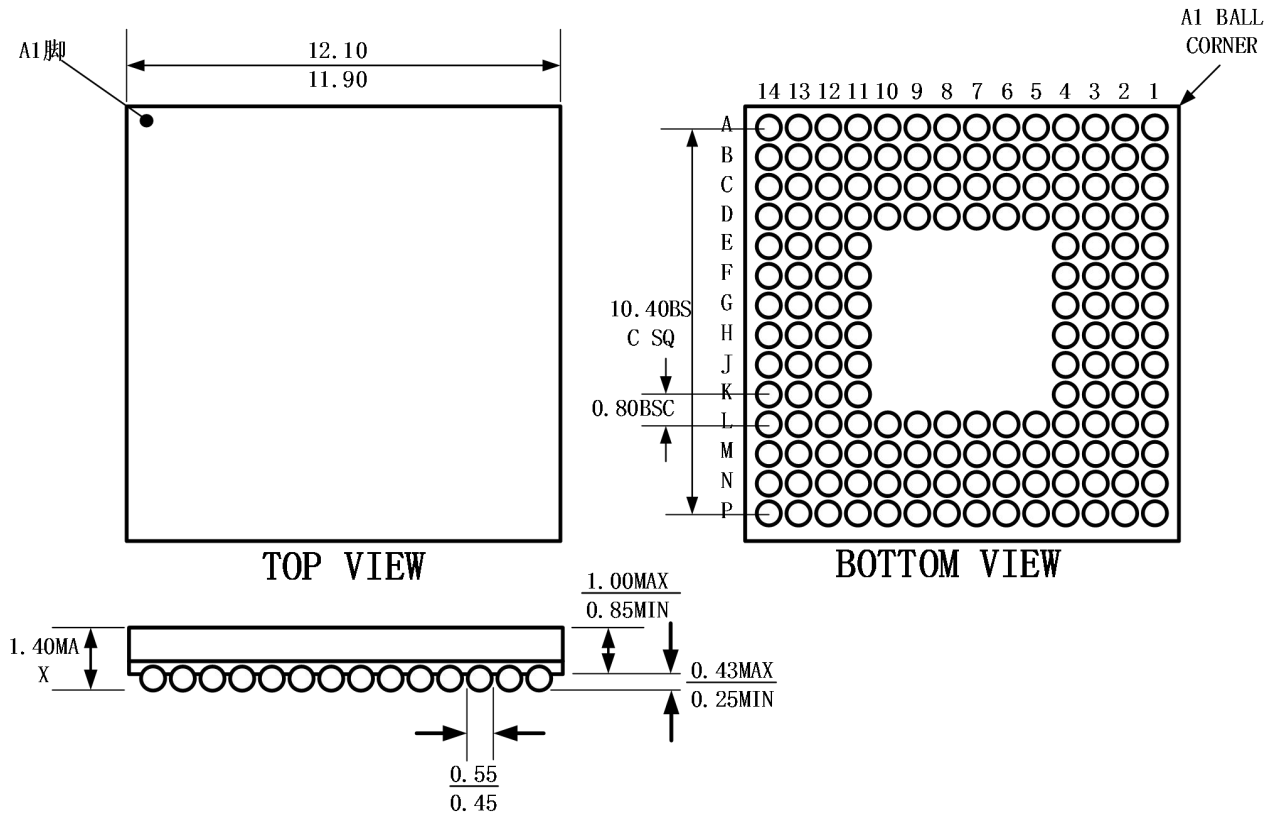
6	0x24	0x30	配置微延迟单元控制器, 在不同的主时钟频率下微延迟单元目标斜率和目标相位值的设定要参考表 3。
7	0x25	0x80	
8	0x27	0x4C	
9	0x28	0x6C	
10	0x29	0xCB	
11	0x26	0x02	
12	0x26	0x03	使能微延迟单元控制器的查找和跟踪模式。
13	未应用	未应用	等待 160k 个主时钟周期。
14	0x2A	--	回读 0x2A 寄存器, 确信它的值为 0x01 以保证 DLL 环路是锁定的。如果未锁定, 则回到步骤 10 并重复操作。如此尝试了 3 次仍不锁定, 就断开环路并报告微延迟单元失败。
15	未应用	未应用	确保从数据源送 DCI 信号到 CBM97D39BG
16	0x13	0x72	设置 FINF_DEL_SKEW 为 2
17	0x10	0x00	关闭数据接收控制器
18	0x10	0x02	使能数据接收控制器环路和中断 (IRQ) 请求
19	0x10	0x03	使能数据接收控制器搜寻和跟踪模式
20	未应用	未应用	等待 135k 个主时钟周期。
21	0x21	--	回读 0x21 寄存器并确定是否为 0x09, 确保 DLL 环路锁定并处于跟踪状态。如果没有锁定和跟踪, 增加寄存器 0x14 的 bit[7:6] (CLKDIVPH[1:0]) 的相位值, 然后从 17 步开始重复操作。最多不超过 3 次尝试, 如果仍然失败后退出环路搜索并报告接收数据锁定失败。
22	0x06, 0x07	0x00, 0x02	可选: 更改 TxDAC 的 IOUTFS 设置 (缺省为 20mA)
23	0x08	0x00	可选: 更改 TxDAC 工作模式 (缺省为正常模式)

表 29 推荐的 SPI 初始化流程 (SYNC 打开)

步骤	寄存器地址 (Hex)	寄存器值	注释
1	0x00	0x00	配置为4线SPI模式, MSB方式。由于上电后MSB/LSB方式未知, 因此位[7:5]必须为位[2:0]的镜像。
2	0x00	0x20	软件复位到SPI的默认值
3	0x00	0x00	清除复位
4	0x22	0x0F	设置输入差分时钟 DACCLK_P 和 DACCLK_N 的共模电压。
5	0x23	0x0F	
6	0x24	0x30	配置微延迟单元控制器, 在不同的主时钟频率下微延迟单元目标斜率和目标相位值的设定要参考表 3。
7	0x25	0x80	
8	0x27	0x4C	
9	0x28	0x6C	
10	0x29	0xCB	
11	0x26	0x02	
12	0x26	0x03	使能Mu控制器搜索和跟踪模式
13	未应用	未应用	等待 160k 个主时钟周期。
14	0x2A	--	回读 0x2A 寄存器, 确信它的值为 0x01 以保证 DLL 环路是锁定的。如果未锁定, 则回到步骤 10 并重复操作。如此尝试了 3 次仍不锁定, 就断开环路 并报告微延迟单元失败。
15	0x15	0x42	配置同步 (SYNC) 控制器
16	0x10	0x00	关闭同步 (SYNC) 控制器
17	0x10	0x60或0x40	使能SYNC控制器环路和中断请求, 0x60 = 主模式; 0x40 = 从模式
18	0x10	0x70或0x50	使能SYNC控制器, 0x70 = 主模式; 0x50 = 从模式
19	未应用	未应用	等待 $160,000 \times 1/f_{DATA}$ 周期, 便于DLL锁定

20	0x21	--	回读0x21寄存器以确定正确工作模式: 0x90 = 主模式; 0x00 = 从模式。如果不为上述值, 需要从第15步开始重复。不超过3次重复操作仍然失败的情况下, 退出环路并上报SYNC锁定失败
21	0x0D	--	回读0x0D寄存器, 确认位[5:4] = 10。如果不等于该值, 回到第二步重新开始操作。不超过3次重复操作仍然失败的情况下, 退出环路并上报SYNC锁定失败
22	未应用	未应用	确保从数据源送DCI信号到CBM97D39BG
23	0x13	0x72	设置窗口宽度为2
24	0x10	0x00	在使能数据接收控制器以前, 先不使能数据接收控制器。
25	0x10	0x02	使能数据接收控制器以建立环路和中断响应。
26	0x10	0x03	使能数据接收控制器搜寻和跟踪模式
27	--	--	等待 $135,000 \times 1/f_{DATA}$ 周期
28	0x21	--	回读0x21寄存器并确定是否为0x09, 确保DLL环路锁定并处于跟踪状态。如果没有锁定和跟踪, 增加寄存器0x14的bit[7:6] (CLKDIVPH[1:0]) 的相位值, 然后从17步开始重复操作。最多不超过3次尝试, 如果仍然失败后退出环路搜索并报告接收数据锁定失败。
29	未应用	未应用	回读主、从芯片中寄存器 0x1B 和 0x1C 中 DCI_DEL 的值, 如果主、从芯片的回读值差异超过了 40 个码值, 则需要重新指定 DCI_DEL 的目标值为主芯片和回读的 DCL_DEL 值的平均值。
30	0x06, 0x07	0x00, 0x02	可选: 更改TxDAC的满幅度输出电流设置 (默认为20mA)
31	0x08	0x00	可选: 更改TxDAC工作模式 (默认为正常模式)

封装尺寸



标注：以上尺寸单位为毫米 (mm)

图 48 CBM97D39BG 外形封装尺寸图

包装/订购信息

产品型号	温度范围	产品封装	丝印	包装数量
CBM97D39AG	-40°C ~ 85°C	PBGA-160	CBM97D39	托盘, 189
CBM97D39BG	-40°C ~ 85°C	PBGA-160	CBM97D39	托盘, 189