

LPM宏模块用法

6.1.1 计数器LPM模块文本代码的调用

(1) 打开LPM宏功能块调用管理器。

*	The MegaWizard Plug-In Manager helps you create or modify design files that contain custom variations of megafunctions. Which action do you want to perform?
\mathbf{A}	Create a new custom megafunction variation
- A 1	C Edit an existing custom megafunction variation
	Copy an existing custom megafunction variation
	Copyright /1551-2005 Altera Corporation

图6-1 定制新的宏功能块

6.1.1 计数器LPM模块文本代码的调用

(1) 打开LPM宏功能块调用管理器。

Which megafunction would you like to cur select a megafunction from the list below	stomize?	Which device family will you be using?	Cydone IV E
Q	×	Which type of output file do you want t	o create?
ALTMULT_ADD ALTMULT_COMPLEX ALTSQRT LPM_ADD_SUB	^	AHDL VHDL Verilog HDL What name do you want for the output	file?
LPM_COMPARE		D:\LPM_MD\CNT4B	
		Output files will be generated using the Return to this page for another creater	dassic file structu ate operation

图6-2 LPM宏功能块设定

6.1.1 计数器LPM模块文本代码的调用

(2) 单击Next按钮后打开如 图6-3所示的对话框

🤣 LP	M_COU	NTER			About Documentation
1 Parameter Settings	2 EDA 3	Summary Optional Inputs			
CNT	4B updown q(3.0)→	Cu	rrently selected	I device family:	Cyclone IV E -
	-1	How wide should the 'q' output bus be? What should the counter direction be?	4 •	bits	
		 Up only Down only Create an 'updown' input port to a 	low me to do b	oth (1 counts up	o; 0 counts down)

图6-3 设4位可加减计数器

6.1.1 计数器LPM模块文本代码的调用

(3) 再单击Next按钮,打开如图6-4所示的对话框



图6-4 设定计数器,含时钟使能和进位输出

6.1.1 计数器LPM模块文本代码的调用

(4) 再单击Next按钮,打开如图6-5所示的对话框



图6-5 加入4位并行数据预置功能

6.1.2 LPM计数器代码与参数传递语句应用

【例 6-1】

module CNT4B (aclr, clk en, clock, data, sload, updown, cout, q); input aclr, clk en; // 异步清 0, 1清 0; 时钟使能, 1 使能, 0 禁止 input clock, sload; // 时钟输入; 同步预置数加载控制, 1 加载, 0 计数 input [3:0] data; input updown; //4 位预置数和加减控制, 1 加, 0 减 output cout; output [3:0] q; // 进位输出和 // 4 位计数输出 wire sub wire0; wire [3:0] sub wire1; // 定义内部连线 wire cout = sub wire0; // 与 assign相同的赋值语句 wire [3:0] q = sub_wire1[3:0]; // 与 assign 相同的赋值语句 lpm_counter lpm_counter_component(//注意例化语句中未用端口必须接上指定电平 .sload(sload), .clk en(clk en), .aclr(aclr), .data(data), .clock(clock), .updown(updown), .cout(sub_wire0), .q(sub_wire1), .aload(1'b0), .aset(1'b0), .cin(1'b1), .cnt en(1'b1), .eq(), .sclr(1'b0), .sset(1'b0));

defparam

6.1.2 LPM计数器代码与参数传递语句应用

【例 6-2】

module REG24B (d, clk, q); input [23:0] d; input clk; output [23:0] q; lpm_ff U1(.q (q[11:0]), .data (d[11:0]), .clock (clk)); defparam U1.lpm_width = 12; lpm_ff U2(.q(q[23:12]), .data(d[23:12]), .clock(clk)); defparam U2.lpm_width = 12; endmodule

【例 6-3】

module CNT4BIT (RST,ENA,CLK,DIN,SLD,UD,COUT,DOUT); input RST, ENA, CLK, SLD,UD ; input [3:0] DIN; output COUT; output [3:0] DOUT ; CNT4B U1(.sload (SLD), .clk_en (ENA), .aclr (RST), .cout (COUT), .clock (CLK), .data (DIN), .updown (UD), .q (DOUT)); endmodule

6.1.3 创建工程与仿真测试



图6-6 CNT4BIT.v的仿真波形

【例 6-4】

module MULT8 (A1, B1, A2, B2, R1, R2) ;

output signed[15:0] R1, R2 ; // 定义有符号数据类型输出 input signed[7:0] A1,B1,A2,B2; // 定义有符号数据类型输入 wire [15:0] R2 /* synthesis multstyle = "logic" */; wire [15:0] R1 /* synthesis multstyle = "dsp" */; assign R1 = A1 * B1 ; assign R2 = A2 * B2 ; endmodule

Flow Status	Successful - Tue May 23 21:32:59 2017
Quartus II 64-Bit Version	13.1.0 Build 162 10/23/2013 SJ Full Versio
Revision Name	MULT8
Top-level Entity Name	MULT8
Family	Cyclone IV E
Device	EP4CE55F23C8
Timing Models	Final
Total logic elements	190 / 55,856 (< 1 %)
Total combinational functions	190 / 55,856 (< 1 %)
Dedicated logic registers	0 / 55,856 (0 %)
Total registers	0
Total pins	64 / 325 (20 %)
Total virtual pins	0
Total memory bits	0 / 2,396,160 (0 %)
Embedded Multiplier 9-bit elements	0/308(0%)
Total PLLs	0/4(0%)

图6-7 完全用逻辑宏单元构建乘法器的编译报告

Flow Status	Successful - Tue May 23 21:30:42 2017
Quartus II 64-Bit Version	13.1.0 Build 162 10/23/2013 SJ Full Versio
Revision Name	MULT8
Top-level Entity Name	MULT8
Family	Cydone IV E
Device	EP4CE55F23C8
Timing Models	Final
Total logic elements	0 / 55,856 (0%)
Total combinational functions	0 / 55,856 (0%)
Dedicated logic registers	0 / 55,856 (0%)
Total registers	0
Total pins	64/325(20%)
Total virtual pins	0
Total memory bits	0 / 2,396,160 (0 %)
Embedded Multiplier 9-bit elements	2/308(<1%)
Total PLLs	0/4(0%)

图6-8 调用了DSP模块的编译报告

	(-	
ategory:		More Analysis & Synthesis Settings			
General	Analysis & Synthe	Specify the settings for the logic options in your project	Assignments made to an individ	ual node	
Files Libraries	Specify options for a	or entity in the Assignment Editor will override the optio	n settings in this dialog box.		
 Operating Settings and Conditions 	affect VQM or EDIF	Eviction ontion eattings:			
Voltage Temperature	Optimization Techr	Name-	Setting:		
 Compilation Process Settings 	Speed	Auto ROM Replacement	On		
Early Timing Estimate	Balanced	Auto Resource Sharing	Off		
Physical Synthesis Optimizations	C Area	Auto Shift Register Replacement	Auto		
EDA Tool Settings	0.000	Block Design Naming	Auto		
Design Entry/Synthesis	Timing-Driven Sy	Carry Chain Length	70	-	
Simulation		Clock MUX Protection	On		
Formal Verification	Power-Up Don t	Create Debugging Nodes for IP Cores	Off		
Analysis & Synthesis Settings	Perform WYSIW	DSP Block Balancing	DSP blocks		
VHDL Input	PowerPlay power on	Disable Register Merging Across Hierarchies	Auto	=	
Verilog HDL Input	romanay pone op	Extract VHDL State Machines	On	-	
Default Parameters	More Settings	Extract Verilog State Machines	On		

图6-9 选择DSP Block Balancing为DSP blocks

6.3.1 初始化文件及其生成

1..mif格式文件

(1) 直接编辑法。

(2) 文件直接编辑法。

(3) 高级语言生成。

(4) 专用.mif文件生成器。

Addr	+0	+1	+2	+3	+4	+5	+6	+7
00	80	86	8C	92	98	9E	A5	AA
08	80	B6	BC	C1	C6	CB	DO	D5
10	DA	DE	E2	E6	EA	ED	FO	F3
18	FS	F8	FA	FB	FD	FE	FE	FF
20	FF	FF	FE	FE	FD	FB	FA	F8
28	F5	F3	FO	ED	EA	E6	E2	DE
30	DA	D5	DO	CB	C6	C1	BC	86
38	80	AA	A5	9E	98	92	8C	86
40	7F	79	73	60	67	61	5A	55
48	4F	49	43	3E	39	34	2F	2A
50	25	21	1D	19	15	12	0F	0C
58	0A	07	05	04	02	01	01	00
60	00	00	01	01	02	04	05	07
68	0A	0C	OF	12	15	19	1D	21
70	25	2A	2F	34	39	38	43	49
78	4F	55	5A	61	67	6D	73	79

图 6-10 mif 文件编辑窗

6.3.1 初始化文件及其生成

1...mif格式文件

【例 6-5】 ↔

DEPTH=128; →	
WIDTH=8; ·····	->
ADDRESS_RADIX = 'HEX; ·	\rightarrow
DATA_RADIX = HEX; ·····	\rightarrow
CONTENT	->
BEGIN	-+
•;0800 ······ : ···· 0080;+/	
0001	
0002 ·····: ·····008C;+	
・・・(数据略去)↩	
007E ·····: ····0073;↔	
007F ·····: ····0079;↔	
END;	

//数据深度,即存储的数据个数→ //输出数据宽度↔ //地址数据类型, HEX表示选择十六进制数据类型。

- //存储数据类型,HEX表示选择十六进制数据类型。 //此为关键词↔
- //此为关键词↔

6.3.1 初始化文件及其生成

1..mif格式文件



DATA	7X8. mi	1 - 记事	本
文件(图)	编辑(E)	格式(0)	3
DEPTH	= 128;		
WIDTH	= 8;		
ADDRES	S_RADIX	= HEX;	;
DATA_R	ADIX =	HEX;	
CONTEN	T BEGI	N	
0000 :	0080;		
0001 :	0086;		
0002 :	008C;		
0003 :	0092;		
0004 :	0098;		
0005 :	009E;		
0006 :	00A5;		
0007 :	00AA;		
0008 :	00B0;		
007E :	0073;		
007F :	0079;		
END :			

图6-11 利用mif生成器生成.mif正弦波文件

图6-12 打开.mif文件

6.3.1 初始化文件及其生成

2. .hex格式文件

3. .dat格式文件

6.3.2 以原理图方式对LPM_RAM进行调用

Which megafunction would you like to customize? Select a megafunction from the list below	Which device family will you be using?	Cydone IV E
ALTUFM_I2C ALTUFM_NONE ALTUFM_NONE ALTUFM_PARALLEL ALTUFM_SPI FIFO FIFO partitioner LPM_SHIFTREG RAM initializer RAM: 1-PORT RAM: 2-PORT RAM: 1-PORT	 Which type of output file do you want to AHDL VHDL Verilog HDL What name do you want for the output D:/LPM_MD/RAM1P Output files will be generated using the Return to this page for another creat Note: To compile a project successfully if files must be in the project directory, in the Options dialog box (Tools menu), or of the Settings dialog box (Assignments) 	file?

图6-13 调用单口LPM RAM

6.3.2 以原理图方式对LPM_RAM进行调用

Parameter 2 EDA 3 Summary				
Widths/Bik Type/Ciks > Regs/Ciken/Byte Er	vable/Adrs $>$ Read During Write Option $>$ Mem	Init	>	
RAM1P	Currently selected device family:	Cydo	ne IV E	÷
		💟 Ma	tch project/d	efault
address[60]	How wide should the 'q' output bus be?		8	bits
- <u>8</u>	How many 8-bit words of memory?		128	word
butclock	Note: You could enter arbitrary values for width and de What should the memory block type be?	pth		
Divertype noto	Auto O MLAB	© M	9К	
	🔿 M14K 💮 LCs		Options	
	Set the maximum block depth to Auto - v	words		
	What clocking method would you like to use?			
	Single dock			
	Dual clock: use separate 'input' and 'output' clocks	C		

图6-14 设定RAM参数

6.3.2 以原理图方式对LPM_RAM进行调用



图6-15 设定RAM仅输入时钟控制



图6-16 设定在写入同时读出原数据: Old Data

6.3.2 以原理图方式对LPM_RAM进行调用



图6-17 设定初始化文件和允许在系统编辑

6.3.2 以原理图方式对LPM_RAM进行调用



图6-18 在原理图上连接好的RAM模块

6.3.3 测试LPM_RAM



图6-19 图6-18的RAM仿真波形

6.3.4 Verilog代码描述的存储器初始化文件加载表述

/* synthesis ram_init_file="DATA7X8.mif" */ ;

(* ram_init_file = "DATA7X8.mif" *) reg[7:0] mem[127:0]

【例 6-6】

```
module RAM78 (output[7:0]Q, input[7:0]D, input[6:0]A, input CLK, WREN);
reg[7:0] mem[0:127] ;
always @(posedge CLK ) if (WREN) mem[A] <= D;
assign Q = mem[A];
initial $readmemh("RAM78_DAT.dat", mem );
endmodule
```

6.3.5 存储器设计的结构控制



图6-20 例6-6的RTL电路模块图

6.3.5 存储器设计的结构控制

【例 6-7】

```
module RAM78(output reg[7:0] Q, input[7:0] D, input[6:0] A, input CLK, WREN);
    reg[7:0] mem[127:0] /* synthesis ram_init_file="DATA7X8.mif" */;
    always @(posedge CLK ) if (WREN) mem[A] <= D;
    always @(posedge CLK ) Q = mem[A];
    endmodule</pre>
```



图6-21 例6-7的RTL电路模块图

6.4.1 简易正弦信号发生器设计

ROM: 1-PORT	About Documenta	ation
Parameter Image: Constraint of the sector	nt >	
address[60] addres	Do you want to specify the initial content of the memory? No, leave it blank Initialize memory content data to XXX on power-up in simulation Yes, use this file for the memory content data (You can use a Hexadecimal (Intel-format) File [.hex] or a Memory Initialization File [.mif]) Browse	
	File name: ./DATA7X8.mif The initial content file should conform to which port's dimensions? Image: Allow In-System Memory Content Editor to capture and	*
	The 'Instance ID' of this ROM is: ROM8	

图6-22 加入初始化配置文件并允许在系统访问ROM内容

6.4.1 简易正弦信号发生器设计



图6-23 正弦信号发生器结构框图

6.4.1 简易正弦信号发生器设计



图6-24 正弦信号发生器电路原理图

6.4.1 简易正弦信号发生器设计



图6-25 图24电路仿真波形

6.4.2 正弦信号发生器硬件实现和测试

🛂 SignalTap II L	ogic Analyzer	- D:/LPM_MD/SI	N_GNT - SIN_GNT -	[output_files/S	IN_stp1.stp]*
File Edit View	Project Pro	cessing Tools V	Andow Help 💖		
Instance Manager	- 🔊 🔊 🔳	Ready to a	cquire	😡 ×	JTAG Chain Configuration: JTAG ready
Instance SNT1	Status Not running	LEs: 690 690 cells	Memory: 61440 61440 bits	Small: 0/0 0 blocks	Hardware: USB-Blaster [USB-0] Device: @1: EP3C55/EP4CE55 (0x020F50DD) >> SOF Manager: . D:/LPM_MD/output_files/SI
log: 2017/05/23	23:41:16 Name 150 AR 041	151 152 153) 05h 06h () 98h 96h (154 155 156 07h 08h 09h 0	157 158 Ah <u>(08h) 00</u>	click to insert time 159 160 161 162 163 164 165 166 16 h <u>ODh</u> <u>OEh</u> <u>OFh</u> <u>10h</u> <u>11h</u> <u>12h</u> <u>13h</u> <u>14h</u> h <u>CBh</u> <u>CBh</u> <u>DBh</u> <u>DBh</u> <u>DBh</u> <u>CBh</u>

图6-26 正弦信号发生器数据输出的SignalTap II实时测试界面

6.4.2 正弦信号发生器硬件实现和测试



图6-27 正弦信号发生器的SignalTap II的波形显示图面

6.5 在系统存储器数据读写编辑器应用

(1) 打开在系统存储单元编辑窗口。

																												Searc	n anera	100m	-
nstance Ma	anage	n ß	a) (-		N	R	eady	to ac	quire							T		×		JTAG	i Chai	n Co	nfigu	ratio	n: J	TAG ready				
Index	Instance ID Status Width De						De	oth	1		T		Hardware:			ardware: US8-Blaster [US8-0]					• Set		é								
0		ş	2M78	i.			Notr	runnir	ŋ		8				128	3			RA		Devia	e:		1: EP	3055	5/EP4	CE55 (0x020F5	(DDD)	•	Scan Cha	sin
č(m			-				-						•		File:	4							1.00		
tance 0: i	RM78	Π.,													_			_													
00000	80	86	80	92	98	9E	A5	AA	B0	B6	BC	C1	C6	CB	DO	D 5	DA	DE	E2	E6	EA	ED	FO	F3	F5	F8					
0001a	FA	FB	FD	FE	FE	FF	FF	FF	FE	FE	FD	FB	FA	FB	F5	F3	FO	ED	EA	E6	E2	DE	DA	D5	DO	CB					
	CG	CI	BC	86	80	AA	AS	98	98	92	80	86	71	79	73	6D	67	61	SA.	55	41	49	43	3E	39	34			.yamq	aZUOIC	1
00034																															

图6-28 In-System Memory Content Editor编辑窗,从FPGA中的ROM读取波形数据

(2) 读取ROM中的数据。

6.5 在系统存储器数据读写编辑器应用

(3) 写数据。

Instance 0: F	RM78																									
000000	11	11	11	11	11	11	11	AA	BO	B6	BC	C1	C6	CB	DO	D5	DA	DE	E2	E6	EA	ED	FO	F3	F5	F8
00001a	FA	FB	FD	FE	FE	FF	FF	FF	FE	FE	FD	FB	FA	F8	F5	F3	FO	ED	EA	E6	E2	DE	DA	D5	DO	CB
000034	C6.	C1	BC	B6	BO	AA	A5	9E	98	92	80	86	7F	79	73	6D	67	61	5A	55	4F	49	43	3E	39	34
00004e	2F	2A	25	21	1D	19	15	12	OF	OC	AO	07	05	04	02	01	01	00	00	00	01	01	02	04	0.5	07
000068	OA	0C	OF	12	15	19	1D	21	25	2A	2F	34	39	3E	43	49	4F	55	5A	61	67	6D	73	79		4

图6-29 在此将编辑好的数据载入FPGA中的ROM内



图6-30 SignalTap II测得的数据波形

(4) 输入输出数据文件。

6.6.1 建立嵌入式锁相环元件

Which megafunction would you like to customize?	Which device family will you be using?	Cydone IV E 👻
	Which type of output file do you want t	o create?
	© VHDL	
ALTPLL ALTPLL_RECONFIG	Verilog HDL What name do you want for the output	file?
ALTREMOTE_UPDATE	D:/LPM_MD/PLL20M	

图6-31 选择锁相环ALTPLL

6.6.1 建立嵌入式锁相环元件

AL	TPLL					About Documentation
1 Parameter Settings	2 PLL Reconfiguration	3 Output Cocks	₫ ÐA	5 Summary		
General/Modes	> Inputs/Lock	> Bandwidt	h∕ss >	Clock switchover	$\boldsymbol{\succ}$	
	PLL20M		1		Currently selected device fa	amily: Cyclone IV E
incik0 incl areset Op	k0 frequency: 20.000 MHz eration Mode: Normal	locked	Abi	e to implement the r eneral	equested PLL	 Exercise to observations.
	1/1 0.00 50.00	Cyclone IV E	v C V	Which device speed	grade will you be using? 8 erature range devices only cy of the inclk0 input? 20	• MHz •

图6-32 选择输入参考时钟inclk0为20MHz

6.6.1 建立嵌入式锁相环元件

PLL20M	Able to implement the requested PLL
inclk0 areset pfdena Clk Ratio Ph (dg) DC (%) c0 1/1 0.00 50.00	Optional Inputs Create an 'plena' input to selectively enable the PLL Create an 'areset' input to asynchronously reset the PLL Create an 'pfdena' input to selectively enable the phase/frequency detector
Cyclone IV E	Lock Output Create 'locked' output Enable self-reset on loss lock

图 6-33 选择 → 锁相环的控制信号

6.6.1 建立嵌入式锁相环元件

PLL20M	c0 - Core/External Output C Able to implement the requested PLL	lock			
incik0 incik0 frequency: 20.000 MHz C0	Use this clock Clock Tap Settings	Requested	Setting	р	Actual Settings
Cik Ratio Ph (dg) DC (%)	Enter output clock frequency:	0.002		MHz 🔻	0.002000
e0 1/10000 0.00 50.00	Enter output dock parameters: Clock multiplication factor	1	(8)		1
Cyclone IV E	Clock division factor	1	1	<< Copy	10000
	Clock phase shift	0.00	0	deg 🔻	0.00
	Clock duty cycle (%)	50.00	٢	1	50.00

图6-34 选择c0的输出频率为0.002MHz

6.6.1 建立嵌入式锁相环元件

in all 0	PLL20M		c1 - Core/External Output Cle Able to implement the requested PLL	ock		
Inciku	inclk0 frequency: 20.000 MHz Operation Mode: Normal		Clock Tap Settings O Enter output clock frequency:	Requested S	ettings	Actual Settings
	Clk Ratio Ph (dg) DC (%) c0 1/10000 0.00 50.00 c1 39/4 0.00 50.00		 Enter output dock parameters: Clock multiplication factor 	1		39
	Cyclone IV	E	Clock division factor	1	<< Copy	4
-			Clock phase shift	0.00	🔹 deg 🔻	0.00

图6-35 输出第二个时钟信号c1

6.6.1 建立嵌入式锁相环元件



图6-36 采用了嵌入式锁相环作时钟的正弦信号发生器电路

6.6.2 测试锁相环

6.7 In-System Sources and Probes Editor用法

(1) 在顶层设计中嵌入In-System Sources and Probes模块。

(2) 设定参数。

a In-System Source	s and Probes	About D	cumentation
1 Parameter Settings 2 EDA 3 Summary			
JTAG1	Currently selected device family:	Cyclone III	
probe[150]	Do you want to specify an Instance Index? No, assign it automatically Yes, use this number	V Match proje	ct/default
	The 'Instance ID' of this instance (optional): JGTT		4 characters
	How wide should the probe port be? 16	*	bits
	How wide should the source port be? 3 Advanced Options	•	bits

图6-37 为In-System Sources and Probes模块设置参数

6.7 In-System Sources and Probes Editor用法

(3) 与需要测试的电路系统连接好。



图6-38 在电路中加入In-System Sources and Probes测试模块

6.7 In-System Sources and Probes Editor用法

(4) 调用In-System Sources and Probes Editor。

File Ed	it View	Processing	Tools W	ndowr He	* *)											
Instance	Manager	- M M		Rea	dv to ac	ouire					0	×	JTAG Chain	Configura	tion: JTA	6 ready	
Probe	read inter	val	01100	Cano -	Eventio	9							Hardware:	USB-Blas	ter [US8-0]		
Currer	t interval	: 0 samples per s	econd		Maximu	m size:	16	•					Device:	@1: EP3	C 16/EP-4CE	15 (0x020	F20DD)
O AL	tomatic				Sav	e data to eve	ent log						File:	D:APM N	MD1/SIN G	NT.sof	
O Ma	nual 1		s	-									(silen o			
-				v	inte sou	rce data:	ontinuou	\$γ ▼] [1.									
1	ndex	Instance I	D	Status		Sources: 3	5	Probes: 10		N	lame	- 11					
2 10 10	π										6 (178)						
Index	Туре	k Name	Data	i ⁻¹⁶	-15	-14		-13	-12		-11	-10		9	-8	-7	
P15	**	COUT	0	Ì	-									2			
[14_8]	**	⊞ AR[60]	37h	27h		28h)	29h	2/	h_)(28h	2Ch		2Dh	2Eh	1 3	Fh (30h
P[70]	**	⊞ Q[70]	BCh	FAh		F8h)	FSh	F3	<u>h (</u>	F0h	EDh	X	EAh	E6h) E	2h 🔾	DEh
		RST	1	1													
52				_													
52 51		EN	1														

图6-39 In-System Sources and Probes Editor的测试情况

6.8.1 DDS原理

$$S_{\rm out} = A\sin\omega t = A\sin(2\pi f_{\rm out}t)$$
 (6-1)

$$\theta = 2\pi f_{out} t \tag{6-2}$$

$$\Delta \theta = 2\pi f_{\text{out}} T_{\text{elk}} = \frac{2\pi f_{\text{out}}}{f_{\text{elk}}}$$
(6-3)

$$\frac{B_{\Delta\theta}}{2^{N}} = \frac{f_{\text{out}}}{f_{\text{elk}}}, \quad B_{\Delta\theta} = 2^{N} \cdot \frac{f_{\text{out}}}{f_{\text{elk}}}$$
(6-4)

6.8.1 DDS原理

$$S_{\text{out}} = A\sin(\theta_{k-1} + \Delta\theta) = A\sin\left[\frac{2\pi}{2^N} \cdot \left(B_{\theta_{k-1}} + B_{\Delta\theta}\right)\right] = Af_{\sin}\left(B_{\theta_{k-1}} + B_{\Delta\theta}\right)$$
(6-5)

$$B_{\theta_{k-1}} \approx \frac{\theta_{k-1}}{2\pi} \cdot 2^N \tag{6-6}$$

$$f_{\rm out} = \frac{B_{\Delta\theta}}{2^N} \cdot f_{\rm clk}$$
 (6-7)

$$f_{\rm out} = \frac{f_{\rm elk}}{2^N}$$
 (6-8)

6.8.1 DDS原理



图6-40 基本DDS结构

6.8.2 DDS信号发生器设计示例



图6-41 DDS信号发生器电路顶层原理图

6.8.2 DDS信号发生器设计示例

$$f_{\rm out} = \frac{B[31..0]}{2^{32}} \cdot f_{\rm elk}$$
 (6-9)



图6-42 图6-41的仿真波形



实验6-1 查表式硬件运算器设计

实验6-2 正弦信号发生器设计

实验6-3 简易逻辑分析仪设计





图6-43 逻辑数据采样电路顶层设计

实验与设计



图6-44 逻辑数据采样电路时序仿真波形



实验6-4 DDS正弦信号发生器设计



图6-45 DDS正弦信号发生器顶层原理图



实验6-5 移相信号发生器设计

实验6-6 AM幅度调制信号发生器设计

实验6-7 硬件消抖动电路设计



【例 6-8】

module ERZP (CLK, KIN, KOUT); //工作时钟和输入信号 input CLK, KIN; output KOUT; req KOUT; req [3:0] KH, KL; always @(posedge CLK) begin if (!KIN) KL<=KL+1 ; else KL<=4'b0000; end //若出现高电平,则计数器清 0 always @(posedge CLK) begin if (KIN) KH<= KH+1; else KH<=4'b0000; end //若出现高电平,则计数器清 0 always@(posedge CLK) begin if (KH > 4'b1100) KOUT<=1'B1;//对高电平脉宽计数一旦大于 12, 则输出 1 else if (KL > 4'b0111) //对低电平脉宽计数若大于 7,则输出 o KOUT<=1'B0; end endmodule

//定义对高电平和低电平脉宽计数之寄存器。

//对键输入的低电平脉宽计数

//同时对键输入的高电平脉宽计数





图6-46 例6-8消抖动电路仿真波形