

基于 FPGA 的相控阵天线波束控制算法优化研究

高 嵩, 胥剑涛

(成都理工大学 信息科学与技术学院(网络安全学院、牛津布鲁克斯学院), 四川 成都 610059)

摘要: 目前 FPGA 是开发波束控制系统的首选控制芯片, 但 FPGA 在计算浮点数据时存在耗时长并且资源占用大的问题, 对波束控制算法的实现存在一定限制。基于此提出一种查表与 FPGA 计算相结合的优化型波束控制算法, 以 8×8 矩形相控阵天线为控制对象, 详细介绍了该优化型算法以及 FPGA 算法模块的逻辑设计, 优化型算法降低 FPGA 资源占用率, 节约系统功耗的同时又规避了 FPGA 难以计算三角函数的问题, 最后基于 Vivado/Xsim 搭建仿真平台, 仿真与测试结果证明了该方案的正确性和高效性。

关键词: FPGA; 波束控制系统; 波束控制算法; 相控阵天线; 三角函数

中图分类号: TN958.92

文献标识码: A

DOI: 10.16157/j.issn.0258-7998.211372

中文引用格式: 高嵩, 胥剑涛. 基于 FPGA 的相控阵天线波束控制算法优化研究[J]. 电子技术应用, 2021, 47(9): 82-85, 95.

英文引用格式: Gao Song, Xu Jiantao. Optimization of phased array antenna beam control algorithm based on FPGA[J]. Application of Electronic Technique, 2021, 47(9): 82-85, 95.

Optimization of phased array antenna beam control algorithm based on FPGA

Gao Song, Xu Jiantao

(School of Information Science and Technology(School of Cyber Security, Oxford Brookes College), Chengdu University of Technology, Chengdu 610059, China)

Abstract: At present, FPGA is the preferred control chip in the development of beam control system, but FPGA has the problems of long time consuming and large resource occupation in computing floating-point data, which limits the implementation of beam control algorithm. Based on this, this paper proposes an optimized beam control algorithm which combines look-up table and FPGA calculation. Taking 8×8 rectangular phased array antenna as the control object, the optimization algorithm is introduced in detail. Finally, a simulation platform is built based on Vivado/Xsim. The simulation and test results prove the correctness and efficiency of the scheme.

Key words: FPGA; beam control system; beam control algorithm; phased array antenna; trigonometric function

0 引言

相控阵天线是相控阵雷达的重要组成部分, 其天线阵面上排列着许多天线阵元^[1]。每一个天线阵元后都配置有移相器来改变阵元通道间的相位关系。在波束控制(下称“波控”)系统的控制下, 改变天线阵元通道之间的幅度和相位关系可以快速地改变天线波束的形状以及波束的指向^[2], 波控系统的优良与否是衡量相控阵性能的重要标准之一^[3-4]。

对于要求安全快速布相的产品来说, 能够快速并行运算的 FPGA 芯片是设计波控系统最好的选择^[5]。但 FPGA 存在难以实现浮点运算的问题^[6], 对于波控算法中三角函数的计算很不方便, 现 FPGA 实现波控算法的方式通常有两种: 一是通过调用 IP core CORDIC 计算的方法; 二是通过查表的方法。以上两种方法中, 方法一在 FPGA 中实现比较困难, 在调用 CORDIC 运算前, 需要

先将输入角度转换为 16 bit 量化的相位值, 且相位值需满足 $\theta \in [-\pi, \pi]$, 并且输出值的小数也难以直接代入公式计算, 这种方法不仅浪费资源而且耗时长。而单一的查表法虽然结构简单, 工作稳定, 波束形成快速, 但如若阵元数目过多或指向角精度要求过高, 需要存储的码值太多, 数据量太大, 也难以实现在大型阵列高指向精度的场合^[7]。张延曹等人使用直接查表法实现了对 16 通道相控阵天线系统的控制^[8]。但单一的查表法只对这种阵元数目少, 指向精度不高的场合适用^[9]。随后郭立俊提出了一种查表法与实时计算相结合的方法, 这种方法虽然规避了 FPGA 难以实现浮点运算的缺陷, 但在设计中生成了多个 ROM 表, 增加了 FPGA 资源占用率的同时也增加了系统功耗。航空航天产品对设备功耗要求很严格, 增加设备的功耗等于降低了设备有效作战时间^[10]。

因此本文以 8×8 矩形相控阵天线为控制对象, 提出

一种查表与 FPGA 计算相结合的优化型算法。由于三角函数的计算对 FPGA 来说比较困难,因此有针对性地在 MATLAB 中生成方向角和俯仰角的三角函数样本,并在 FPGA 算法设计中利用三角函数的诱导公式,减少了 ROM 表数量的同时也缩减了存储空间。系统工作时, FPGA 所需要做的仅仅是根据上位机信号查表取数据做有限的运算,这种优化型方法更节约资源,速度快而且误差也相对小。

1 波束扫描原理

根据阵列天线理论,相邻两个阵元之间接收的信号存在由于路程差引起的相位差^[11]:

$$\phi = \frac{2\pi d \sin\theta}{\lambda} \quad (1)$$

式中: d 为两个天线阵元之间的距离; λ 为信号的波长, θ 为波束方向与两个天线阵元垂直平面之间的夹角^[12]。反之,只需要改变阵内相邻阵元之间的相位差 ϕ ,就可以改变波束的最大指向角 θ ,即实现天线的波束扫描^[13]。

矩形相控阵天线阵面坐标如图1所示,阵面位于 xoy 平面,以方向角和俯仰角来表示目标所在的方向,其中 θ 为俯仰角,范围为 $0^\circ \sim 90^\circ$, φ 为方向角,范围为 $0^\circ \sim 360^\circ$ 。

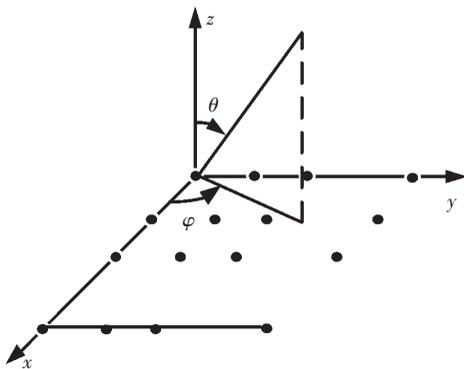


图1 相控阵天线阵面坐标图

图2为坐标的 xoy 平面,矩形天线阵面由 $M \times N$ 个天线阵元组成, d_x 和 d_y 分别是 x 方向和 y 方向的天线阵元间距。

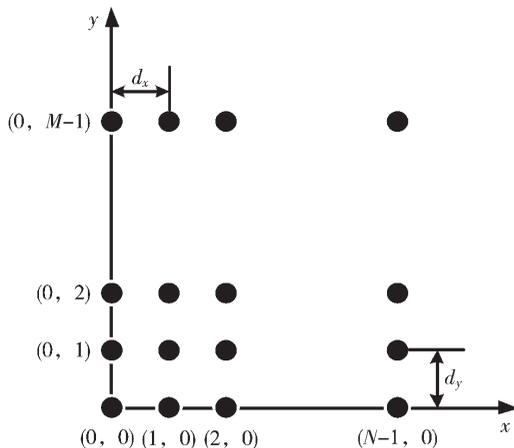


图2 天线 xoy 平面图

阵列天线中,相邻两阵元间的 x 轴方向和 y 轴方向的相位差分别为 β' 和 α' ^[18]。以 (m', n') 阵元为参考阵元时, (m, n) 阵元相对于参考阵元的移相码可由式(2)计算^[14]:

$$C(m, n) = (m - m')\beta' + (n - n')\alpha' \quad (2)$$

其中: $m = 1, 2, \dots, M-1$; $n = 1, 2, \dots, N-1$ 。

$$\beta' = \frac{2\pi d_x \sin\theta \cos\varphi}{\lambda} \quad (3)$$

$$\alpha' = \frac{2\pi d_y \sin\theta \sin\varphi}{\lambda} \quad (4)$$

其中 $\lambda = c/f$, c 为光速, f 为频率。

2 算法实现

图3为 8×8 矩形相控阵天线的 xoy 平面图。

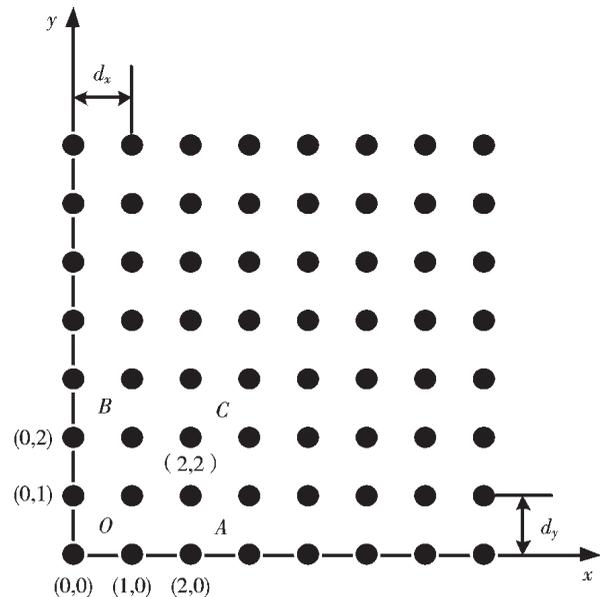


图3 8×8 矩形天线阵面图

以 8×8 矩形相控阵天线为控制对象,简要分析波控算法的计算过程。以左下角点 O 为坐标原点,以原点 $(0, 0)$ 作为式(2)的 (m', n') ,其余通道阵元的移相码都相对于 O 点进行运算,可知:原点天线阵元 $O(0, 0) = 0 \cdot \beta' + 0 \cdot \alpha'$;则天线阵元 $A(2, 0) = 2 \cdot \beta' + 0 \cdot \alpha'$; $B(0, 2) = 0 \cdot \beta' + 2 \cdot \alpha'$; $C(2, 2) = 2 \cdot \beta' + 2 \cdot \alpha'$;显然: $C(2, 2) = A(2, 0) + B(0, 2)$ 。

在计算整个阵列时,带有三角函数的 β' 和 α' 可以通过查表和运算得到, FPGA 只需要简单地计算出 x 轴和 y 轴上面的阵元移相码,其余阵元均可通过对应点相加得到。

2.1 样本生成和算法优化

根据天线阵面要求,对样本生成和 FPGA 算法进行了设计优化。在样本生成前,分析式(3)、式(4)中含有 2π 可知其计算结果为弧度值,而要求的移相码为角度值,因此需要进行单位转换;其次,移相角度范围是 $0^\circ \sim 360^\circ$,而本设计采用的控制芯片为 CETC13 所的 NC-15359C-1418SD 和 NC-15359C-1418SDM,两支路,6位数控移相,因此需要将 360° 量化为6 bit 的移相码,则 $1\text{LSB} = 360/2^6 =$

5.625°。具体单位转化及量化过程如下：

单位转化：由弧度转角度公式“角度值 1°=360/2π”可知，角度值公式如下^[15]：

$$\beta' = \frac{360^\circ \cdot d_x \cdot \sin\theta \cdot \cos\varphi \cdot f}{c} \quad (5)$$

同理，

$$\alpha' = \frac{360^\circ \cdot d_y \cdot \sin\theta \cdot \sin\varphi \cdot f}{c} \quad (6)$$

量化：由 β'(量化值)=β'(角度)/1LSB, 6 bit 量化值公式如下：

$$\beta' = \frac{2^6 \cdot d_x \cdot \sin\theta \cdot \cos\varphi \cdot f}{c} \quad (7)$$

同理，

$$\alpha' = \frac{2^6 \cdot d_y \cdot \sin\theta \cdot \sin\varphi \cdot f}{c} \quad (8)$$

根据量化后的公式，MATLAB 样本文件生成部分代码如图 4 所示。

```
sin_theta =round((2^36)*(2^6)*sin(theta*step*pi/180)/c) ;
sin_phi   =round(dy*sin(theta*step*pi/180)*(2^20)) ;
cos_phi   =round(dx*cos(theta*step*pi/180)*(2^20)) ;
```

图 4 样本生成 MATLAB 部分代码

以图 4 简要介绍采用 MATLAB 生成三角函数样本的流程，俯仰角 θ 以步进 step 0.001 变化，为了减少 FPGA 计算时间和资源，将公式中的定值 c、2⁶ 一并乘入在样本中，之后再再将样本值扩大并四舍五入取整生成 coe 文件，并存入 FPGA 的 ROM1 中。扩大样本值是由于 FPGA 实现小数乘法很困难，无论使用 CORDIC 或者 DSP48E 计算均比较复杂，并且这两种 IP core 输出的小数数据格式也难以直接带入波控公式^[8]。

同理把系统将输入的方向角 φ 也以步进 step 0.001 变化，最后也以整数的形式生成正弦值和余弦值样本存入 ROM 中。但此处存在一个问题，由于方向角 φ 的范围为 0°~360°，若正弦值和余弦值样本都以步进 0.001 变化，ROM 表数据会达到 720 000 个，若指向角精度要求更高，需要存储的码值将会成倍增加。

针对此问题，本文提出的优化算法部分代码如图 5、图 6 所示。

通过三角函数的诱导公式可知，只需要生成第一象限的样本值，便可以得到整个周期的三角函数值。通过

```
else if (fx_angle>=0&&fx_angle<=90_000)begin
    sin_phi <=rom_douta ;
    rom_addr <=fx_angle ;
end else if (fx_angle>=90_000&&fx_angle<=180_000)begin
    sin_phi <=rom_douta ;
    rom_addr <=180_000-fx_angle ;
end else if (fx_angle>=180_000&&fx_angle<=270_000)begin
    sin_phi <=rom_douta * (-1) ;
    rom_addr <=fx_angle-180_000 ;
end else if (fx_angle>=270_000&&fx_angle<=360_000)begin
    sin_phi <=rom_douta * (-1) ;
    rom_addr <=360_000-fx_angle ;
end
```

图 5 优化算法正弦部分代码

```
else if (fx_angle>=0&&fx_angle<=90_000)begin
    cos_phi <=rom_douta ;
    rom_addr <=90_000-fx_angle ;
end else if (fx_angle>=90_000&&fx_angle<=180_000)begin
    cos_phi <=rom_douta * (-1) ;
    rom_addr <=fx_angle-90_000 ;
end else if (fx_angle>=180_000&&fx_angle<=270_000)begin
    cos_phi <=rom_douta * (-1) ;
    rom_addr <=270_000-fx_angle ;
end else if (fx_angle>=270_000&&fx_angle<=360_000)begin
    cos_phi <=rom_douta ;
    rom_addr <=fx_angle-270_000 ;
end
```

图 6 优化算法余弦部分代码

优化算法可以将正弦函数样本和余弦函数样本的数据量都缩减为 1/4 并在查表时共用一个 ROM 表，减少 ROM 表数量的同时数据量也从优化前的 720 000 个缩减到 90 000 个，减少了 FPGA 芯片的 LUT 和 BRAM 资源占用率，降低整个波控系统的功耗。

2.2 FPGA 算法模块逻辑设计

根据优化型算法对算法模块进行了逻辑设计，FPGA 算法模块逻辑设计框图如图 7 所示。

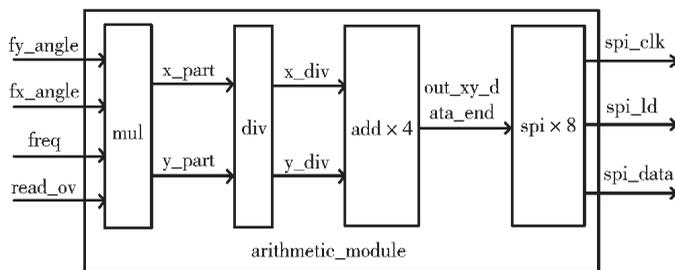


图 7 FPGA 算法模块设计框图

算法模块包括 mul 模块、div 模块、add 模块和 spi 模块。算法模块的输入为从上位机接收到俯仰角、方向角以及频率等参数。mul 模块中调用了 Block Memory Generator ROM IP core 存储三角函数值，mul 模块的作用在于根据角度值进行查表取值，以及做乘法运算得到 β'、α' 和 x 轴、y 轴上的基底值。div 模块用于接收 mul 模块处理好的基底值 x_part 和 y_part，为数据进行除法截位操作，还原在 MATLAB 中的扩大操作并进行四舍五入。add 模块的输入数据为 x_div 和 y_div，该模块的主要功能是将还原后的基底值相加得到各个阵元通道的移相码，在程序中例化 add 模块 4 次，每一个模块计算两行通道的移相码，并行计算 4 个 add 模块，加快移相码计算速度，在得到各通道移相码后，根据数字移相器位数截位截取低 6 bit 送入 spi 模块进行 spi 分发。T/R 组件和波控系统之间采用 spi 接口连接，波控系统为主设备，T/R 组件作为从设备，spi 模块在程序中例化 8 次，产生 8 路 spi 时钟，每路 spi 时钟同时驱动 4 路移相器，加快移相码分发速度。

算法模块具体工作流程如下：波控系统从上位机通过 UART 接收到下发的俯仰角、方向角以及频率值等参

数,收到参数后系统通过查表的方式获得方向角和俯仰角的三角函数值,根据式(3)、式(4)做乘法运算得到 β' 、 α' ,再根据式(2)分别乘上坐标值0~7计算出阵列中 x 轴和 y 轴上的基底值,还原在MATLAB中进行的扩大操作并进行四舍五入后,其他坐标的值可通过 x 、 y 轴上的基底值两两相加得到,当所有天线阵元通道都计算完成后,根据数字移相器位数截位得到各个通道最终的移相码,最后将移相码通过spi按照控制芯片顺序分发给T/R组件即可实现整个天线阵面的移相控制。

3 系统仿真测试

FPGA开发基于Vivado2016.4,利用Vivado自带的仿真工具Xsim进行仿真。设定俯仰角为 45° ,方向角为 30° ,频率值为15.5 GHz。

仿真得到分发的移相码结果和MATLAB计算得到的理论移相码如图8、图9所示。

Name	Value	Name	Value	Name	Value
out_11_data_end[5:0]	0	out_36_data_end[5:0]	30	out_63_data_end[5:0]	4
out_12_data_end[5:0]	52	out_37_data_end[5:0]	18	out_64_data_end[5:0]	56
out_13_data_end[5:0]	41	out_38_data_end[5:0]	6	out_65_data_end[5:0]	44
out_14_data_end[5:0]	29	out_41_data_end[5:0]	3	out_66_data_end[5:0]	33
out_15_data_end[5:0]	17	out_42_data_end[5:0]	55	out_67_data_end[5:0]	21
out_16_data_end[5:0]	6	out_43_data_end[5:0]	44	out_68_data_end[5:0]	9
out_17_data_end[5:0]	58	out_44_data_end[5:0]	32	out_71_data_end[5:0]	7
out_18_data_end[5:0]	46	out_45_data_end[5:0]	20	out_72_data_end[5:0]	59
out_21_data_end[5:0]	44	out_46_data_end[5:0]	9	out_73_data_end[5:0]	48
out_22_data_end[5:0]	32	out_47_data_end[5:0]	61	out_74_data_end[5:0]	36
out_23_data_end[5:0]	21	out_48_data_end[5:0]	49	out_75_data_end[5:0]	24
out_24_data_end[5:0]	9	out_51_data_end[5:0]	47	out_76_data_end[5:0]	13
out_25_data_end[5:0]	61	out_52_data_end[5:0]	35	out_77_data_end[5:0]	59
out_26_data_end[5:0]	50	out_53_data_end[5:0]	24	out_78_data_end[5:0]	53
out_27_data_end[5:0]	38	out_54_data_end[5:0]	12	out_81_data_end[5:0]	50
out_28_data_end[5:0]	26	out_55_data_end[5:0]	0	out_82_data_end[5:0]	38
out_31_data_end[5:0]	24	out_56_data_end[5:0]	53	out_83_data_end[5:0]	27
out_32_data_end[5:0]	12	out_57_data_end[5:0]	41	out_84_data_end[5:0]	15
out_33_data_end[5:0]	1	out_58_data_end[5:0]	29	out_85_data_end[5:0]	3
out_34_data_end[5:0]	53	out_61_data_end[5:0]	27	out_86_data_end[5:0]	56
out_35_data_end[5:0]	41	out_62_data_end[5:0]	15	out_87_data_end[5:0]	44
		out_88_data_end[5:0]	32		

图8 Xsim 仿真结果

	1	2	3	4	5	6	7	8
1	0	52	41	29	17	6	58	46
2	44	32	21	9	61	50	38	26
3	24	12	1	53	41	30	18	6
4	3	55	44	32	20	9	61	49
5	47	35	24	12	0	53	41	29
6	27	15	4	56	44	33	21	9
7	7	59	48	36	24	13	1	53
8	50	38	27	15	3	56	44	32

图9 MATLAB 计算结果

6 bit 数据 out_xy_data_end 为波控系统最终通过 spi 分发给各天线通道的移相码,为了方便分析比较,命名方式设置为可与 MATLAB 矩阵坐标对应。例如:Xsim 中移相码 out_78_data_end 为分发给阵面坐标(6,7)的移相码,对应 MATLAB 生成的矩阵第 7 行第 8 列,其结果为 53。通过对比,Xsim 仿真结果与 MATLAB 计算结果无误差。

图 10 为算法模块顶层仿真图。

波控系统为移相器提供了 32 个独立的移相器控制信号(spi_data),8 个时钟信号(spi_clk)以及 8 个锁存信号(spi_ld),一路 spi 时钟同时驱动控制四路移相器,节约了接口资源,加快了布相速度。如图 10 所示,在 0.205 μs 时,

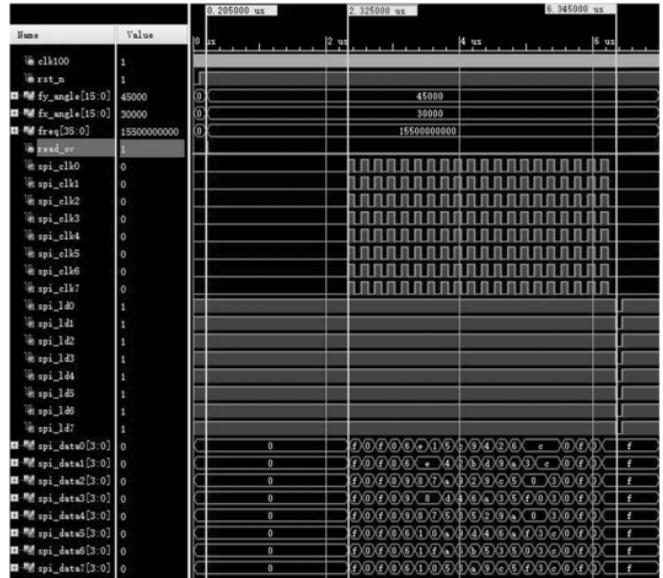


图 10 算法模块仿真图

read_ov 为高,表示系统接收完毕下发的参数;在 2.325 μs 时移相码计算完成开始通过 spi 分发移相码;在 6.345 μs 时,spi_ld 拉低,移相码分发完成,完成了对天线阵面各阵元的控制。将仿真结果与类似文献对比发现^[10],本文所设计的算法模块具有更快的移相码计算速度以及分发速度。仿真结果证明,根据优化型算法所设计的算法模块在保证控制准确性的同时也具有高效的计算速度和布相速度。

4 结论

波控系统是相控阵雷达重要系统之一,在规定时间内完成移相码计算和移相码分发,控制天线波束快速扫描是波控系统主要的功能^[16]。本文运用能快速并行运算的 FPGA 芯片设计波控系统,加快了移相码计算速度和布相速度,采用查表与 FPGA 计算相结合的优化型波控算法,降低系统功耗的同时又解决了 FPGA 难以实现浮点计算的问题。最终仿真结果证明了该方案的正确性和高效性,目前该方案已成功应用于某 KU 宽带数据链相控阵天线系统中。

参考文献

- [1] 许大进,李琳,宣浩.一种机载相控阵雷达波控系统设计与实现[J].雷达科学与技术,2015,13(6):572-576.
- [2] CHEN Z,LIU J,LI L.Design of scalable beam steering system of phased array radar[C]//2011 IEEE CIE International Conference on Radar.IEEE,2011:1153-1156.
- [3] 张光义,赵玉洁.相控阵雷达技术[M].北京:电子工业出版社,2007.
- [4] 古磊,竺小松.有源相控“智能蒙皮”波束控制系统的设计与实现[J].火力与指挥控制,2018,43(4):152-157,164.
- [5] 王鸿雁.基于多波束相控阵天线测量的控制系统设计[D].成都:西南交通大学,2017.
- [6] 钟小艳,苏涛,孙多.FPGA 中高速浮点运算的设计与实现

(下转第 95 页)

Science, 2014, 42(7): 1861-1869.

(收稿日期: 2020-12-30)

作者简介:

袁松(1996-), 男, 硕士研究生, 主要研究方向: 高压等离子电源及智能控制。

曾敏(1970-), 通信作者, 男, 工学博士, 副教授, 主要研究方向: 数字化电源及控制, E-mail: memzeng@scut.edu.cn。

李阳(1966-), 男, 硕士, 讲师, 主要研究方向: 弧焊电源设备及智能控制。



扫码下载电子文档

(上接第 85 页)

现[J]. 火控雷达技术, 2009, 38(2): 48-52.

[7] 尚仁超. 有源相控阵天线波控系统的设计[D]. 南京: 南京理工大学, 2015.

[8] 张延曹, 王勇, 陈灿, 等. 相控阵雷达波束控制器优化设计[J]. 计算机仿真, 2015, 32(11): 51-54.

[9] 车明阳. 基于 FPGA 的相控阵雷达波束控制系统设计[D]. 长春: 长春理工大学, 2014.

[10] 郭立俊. 改进型波控算法在 FPGA 中的研究与实现[J]. 合肥工业大学学报(自然科学版), 2020, 43(2): 200-204.

[11] 何凯. 有源相控阵雷达波控运算单元的研究[D]. 贵阳: 贵州大学, 2018.

[12] 姚志文. 一种弹载相控阵天线波束控制系统设计[J]. 航空兵器, 2016(6): 12-15.

[13] ALBANI M, CADILI T, DI MAGGIO F, et al. A2-D elec-

tronic beam steering phased array for point-multipoint communication application[C]// Radar Conference, 2007. EuRAD, 2007. European. IEEE, 2007: 350-353.

[14] 郭立俊, 姚丹. 一种虚拟化矩形雷达波束控制方法[J]. 雷达科学与技术, 2017, 15(6): 643-646.

[15] 卞子懿. 三角排布有源相控阵天线高精度波控算法设计与实现[D]. 南京: 南京理工大学, 2019.

[16] 张宇驰. 基于 FPGA 平台的波控系统的设计[D]. 南京: 南京理工大学, 2012.

(收稿日期: 2021-02-04)

作者简介:

高嵩(1972-), 男, 博士, 副教授, 主要研究方向: 电磁法地球物理仪器研究。

胥剑涛(1997-), 通信作者, 男, 硕士研究生, 主要研究方向: 地学信号获取及监测处置技术, E-mail: jiantaoxu@126.com。



扫码下载电子文档

(上接第 90 页)

从串口中可以看出单板的 CPU 已启动, 并且 SONY TV 已经被点亮。

最后, 从图 15 得知, 由中间件在短时间内下发的 CEC 消息(不超过 30 条)全部被 CEC 驱动发送了出去, 并且没有一个 CEC 消息发送失败。这表明单板的 CEC 软件设计以及硬件逻辑完全符合该场景下待机唤醒的需求。

4 结论

CPU 驱动提供的 UNF 接口可以让业务指令中心下发所需的 CEC 消息, 配合 MCU 中的驱动, 所有的消息时序符合 CEC1.4 协议中规定, 并且完全满足电视一键待机和一键唤醒的要求。通过 CPU 和 MCU 结合的方式, 也让电视在待机下功耗更低。

参考文献

[1] 黄庆敏, 罗键. HDMI 接口标准及应用设计[J]. 电视技术, 2007, 31(2): 32-34.

[2] 陶晓臻, 张云霄, 于海波. 几种 HDMI 传输方式的比较[J]. 电子世界, 2020(16): 47-48.

[3] 严传高, 张乘浩, 刘马良, 等. HDMI 高速显示数据接口技术[J]. 微纳电子与智能制造, 2020(2): 105-111.

[4] 孙玲, 刘文杰, 施佺, 等. 基于 FPGA 的 HDMI_CEC 设计与实现[J]. 电视技术, 2013, 37(17): 87-90.

[5] 张涌, 黄世震. HDMI CEC 的设计与实现[J]. 中国集成电路, 2010, 19(12): 64-67.

[6] 李新, 梁春明. HDMI 接收端的数据同步模块设计[J]. 电

视技术, 2016, 40(11): 30-34.

[7] 谢特辉. HDMI-CEC 简介及其应用[J]. 电视技术, 2010, 34(10): 42-46.

[8] 孙玲, 刘文杰, 施佺, 等. 基于 FPGA 的 HDMI_CEC 设计与实现[J]. 电视技术, 2013, 37(17): 87-90.

[9] 苗韵, 鲍宜鹏, 杨晓刚. 一种 HDMI-CEC 控制器的设计与实现[J]. 电子与封装, 2020, 20(2): 48-55.

[10] 深圳市智微智能软件开发有限公司. IO 模拟实现 CEC 协议的方法、系统、设备和存储介质: 112000599[P]. 2020-11-27.

[11] 海信视像科技股份有限公司. 显示设备的唤醒方法及显示设备: 111447498[P]. 2020-07-24.

[12] TCL 通力电子有限公司. HDMI-CEC 电路及 HDMI-CEC 设备: 208046769[P]. 2018-11-02.

[13] 陈晓峰. HDMI 接口分析[J]. 中国有线电视, 2013(4): 468-470.

[14] Lionel Belnet. 搭载全新 ARM Cortex-A73 处理器, 智能手机设计能效及性能再获优化[J]. 单片机与嵌入式系统应用, 2016, 16(9): 7-8.

(收稿日期: 2021-03-09)

作者简介:

吕亮(1996-), 男, 硕士研究生, 主要研究方向: 多媒体高速接口。

龙华(1985-), 通信作者, 男, 高级工程师, 主要研究方向: 多媒体高速接口, E-mail: 378492967@qq.com。



扫码下载电子文档

版权声明

经作者授权，本论文版权和信息网络传播权归属于《电子技术应用》杂志，凡未经本刊书面同意任何机构、组织和个人不得擅自复印、汇编、翻译和进行信息网络传播。未经本刊书面同意，禁止一切互联网论文资源平台非法上传、收录本论文。

截至目前，本论文已经授权被中国期刊全文数据库（CNKI）、万方数据知识服务平台、中文科技期刊数据库（维普网）、DOAJ、美国《乌利希期刊指南》、JST 日本科技技术振兴机构数据库等数据库全文收录。

对于违反上述禁止行为并违法使用本论文的机构、组织和个人，本刊将采取一切必要法律行动来维护正当权益。

特此声明！

《电子技术应用》编辑部

中国电子信息产业集团有限公司第六研究所