

引用格式:肖子雅,刘升. 黄金正弦混合原子优化算法[J]. 微电子学与计算机,2019,36(6):21-25. [XIAO Z Y,LIU S. Atom search optimization based on golden-sine algorithm [J]. Microelectroncs & Computer,2019,36(6):21-25.]

黄金正弦混合原子优化算法

肖子雅,刘 升

(上海工程技术大学 管理学院,上海 201620)

摘 要: 原子优化算法是一种基于分子动力学模型的新型智能算法,针对基本原子优化算法收敛速度慢以及易陷入局部最优的局限性,本文将黄金正弦作为局部优化算子嵌入到基本算法中,提出了黄金正弦原子优化算法. 原子个体通过黄金正弦操作可以与最优个体进行充分的信息交流,有效改善基本算法的寻优方式,提高算法的收敛速度和寻优精度. 通过实验表明,改进后的原子优化算法具有更好的寻优性能.

关键词: 原子优化算法;黄金正弦算法;多模态函数

中图分类号: TP3

文献标识码: A

文章编号: 1000-7180(2019)06-0021-05

Atom search optimization based on golden-sine algorithm

XIAO Zi-ya,LIU Sheng

(School of Management , Shanghai University of Engineering Science,Shanghai 201620,China)

Abstract: Atom search optimization is a new intelligent algorithm based on molecular dynamics model. In order to Improve convergence race and overcome the shortcoming of the atom search optimization which is easy to fall into the local extremum and easy to premature convergence ,atom search optimization based on Golden-sine algorithm was proposed. It designs golden sine algorithm as a local optimization operator to optimizing the atom agent. The improved algorithm make each atom agent fully absorb information from the best agent, which effectively improve the optimization way of ASO and improve the convergence and accuracy of the optimization. Experiments show that the improved atom optimization algorithm has better performance.

Key words: atom search optimization;golden sine algorithm;multimodal function

1 引言

原子优化算法 (Atom Search Optimization, ASO)^[1] 是一种基于分子动力学模型^[2] 的新颖智能算法. 模拟在原子构成的分子系统中,原子因相互间的作用力和系统约束力而产生位移的现象. 在一个分子系统中,相邻的原子间存在相互作用力(吸引力和排斥力),且全局最优原子对其他原子存在几何约束作用. 引力促使原子广泛地探索整个搜索空间,斥力使它们能够有效地开发潜在区域. 与花授粉算法^[3]、细菌觅食算法^[4]、蝙蝠算

法^[5]相比,原子优化算法具有更好的收敛精度,但基本的原子优化算法仍存在收敛速度慢且易于陷于局部最优的局限性.

本文针对基本原子搜索算法的局限性进行了改进,提出了一种黄金正弦混合的原子优化算法 (Atom Search Optimization based on Golden-sine Algorithm,GSASO),通过改进提高算法的收敛速度,并减小算法陷入局部停滞的可能性.

2 基本原子优化算法(ASO)

假设一个分子系统是由 s 个原子构成的 d 维空

间, $X_i^d(t)$ 为第 i 个原子在第 t 次迭代时的位置, 可以表示为 $X_i^d(t) = (X_{i,1}, X_{i,2}, \dots, X_{i,d}) (i = 1, 2, \dots, s; t = 1, 2, \dots, t_{\max})$, $t_{\max}$ 为最大迭代次数, $X_{\text{best}}^d(t)$ 为第 t 次迭代时全局最优解。

原子运动遵循经典力学^[6], 根据牛顿第二定律, 原子的加速度与其质量有关, 且由原子间的相互作用力和最优原子对其的几何约束的共同作用产生, 所以第 i 个原子在第 t 次迭代时加速度如下:

$$a_i^d(t) = \frac{F_i^d(t) + G_i^d(t)}{m_i^d(t)} \quad (1)$$

式中, $F_i^d(t)$ 为第 t 次迭代时 d 维空间中作用于第 i 个原子的总力, 可以看作是适应度函数值较好的 k 个原子对第 i 个原子作用力的随机加权之和, 表示如下:

$$F_i^d(t) = \sum_{j \in K_{\text{best}}} \text{rand}_j F_{ij}^d(t) \quad (2)$$

式中, K_{best} 为适应度函数值较好的 k 个原子的集合, $F_{ij}^d(t)$ 为两原子之间的作用势能, 可以表示为:

$$F_{ij}^d(t) = -\eta(t) [2(h_{ij}(t))^3 - (h_{ij}(t))^7] \quad (3)$$

式中, $\eta(t) = \alpha \left(1 - \frac{t-1}{t_{\max}}\right)^3 e^{-\frac{20t}{t_{\max}}}$ 可以调节引力区域和斥力区域的范围, η 随着迭代次数的增加, η 自适应递减, 使得全局搜索和局部开发的范围都逐步缩小至最优值, 保证了算法的收敛性; α 为深度加权; $h_{ij}(t)$ 为两个原子之间的距离, 不同的 h 值对应着不同的作用力性质. 如图 1 所示, 当 $h \in (0.9, 1.12)$ 时为斥力, 且随着 h 值得增大而增大; 当 h 为 1.12 时, 为平衡状态, 作用力为 0; 当 $h \in (1.12, 1.24)$ 时, 为吸引力且随着 h 值得增大而增大, $h \in (1.24, 2)$ 时, 仍为吸引力但随着 h 值得增大而减小至 0, 所以 h 可以表示为:

$$h_{ij}(t) = \begin{cases} h_{\min}, & \frac{r_{ij}(t)}{\sigma(t)} < h_{\min} \\ \frac{r_{ij}(t)}{\sigma(t)}, & h_{\min} \leq \frac{r_{ij}(t)}{\sigma(t)} \leq h_{\max} \\ h_{\max}, & \frac{r_{ij}(t)}{\sigma(t)} > h_{\max} \end{cases} \quad (4)$$

式中, $h_{\min} = \epsilon_0 + \epsilon(t)$ 为 h 的下界; $\epsilon(t)$ 为漂移因子随着迭代次数的变化而变化, 使得算法在全局搜索和局部开发中转换; $h_{\max}$ 为 h 上界; $\sigma(t) =$

$\sum_{j \in K_{\text{best}}} X_{ij}(t) / K(t)$ 是 K_{best} 集合中的原子与第 i 个原子的距离范围。

在 ASO 算法中, 为了加强迭代初期的全局探索能力, 每个原子需要与较多个适应度较好的邻近原

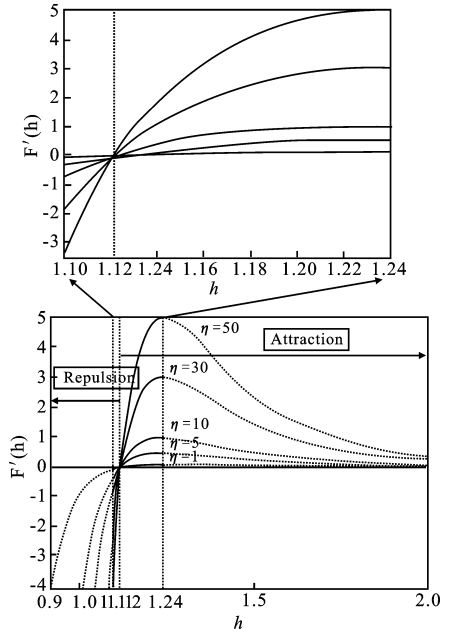


图 1 η 、 F 和 h 关系图

子产生相互作用, 而在迭代后期为了增强局部开发促进算法收敛, 每个原子需要与较少的适应度较好的邻近原子产生相互作用. 适应度较好的邻近原子的

数量用 K 表示, $K = s - (s-2) \times \sqrt{\frac{t}{t_{\max}}}$ 随着迭代次数自适应减小, 既保证了迭代前期算法跳出局部最优进行全局搜索的能力, 又保证了算法后期局部开发能力并保证算法的收敛性. K_{best} 为适应度函数值较好的 k 个原子的集合, 原子群的作用力如图 2 所示。

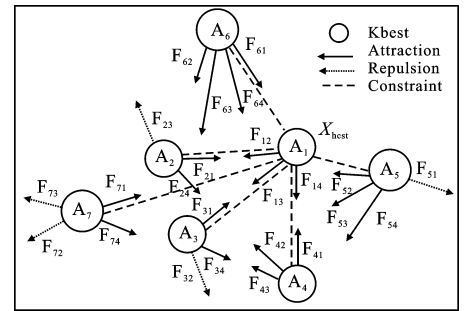


图 2 原子群相互作用示意图($K = 5$)

在分子动力学模型中, 几何约束在原子运动中是十分重要的因素. 在 ASO 中, 为了简单起见, 假设每个原子与最优原子具有共价键, 因此每个原子受来自最佳原子的约束力的作用, 所以 (1) 式中的 $G_i^d(t)$ 为第 t 次迭代时 d 维空间中全局最优原子对第 i 个原子的几何约束作用, 表示为:

$$G_i^d(t) = \lambda(t) (X_{\text{best}}^d(t) - X_i^d(t)) \quad (5)$$

式中, $\lambda(t) = \beta e^{-\frac{20t}{t_{\max}}}$, 随着迭代次数做自适应调整; β 为乘数权重。

(1) 式中 $m_i^d(t)$ 为原子的质量, 可表示为:

$$m_i^d(t) = \frac{M_i(t)}{\sum_{j=1}^N M_j(t)} \quad (6)$$

$$a_i^d(t) = \frac{F_i^d(t)}{m_i^d(t)} + \frac{G_i^d(t)}{m_i^d(t)} =$$

$$-\alpha \left(1 - \frac{t-1}{T}\right)^3 e^{-\frac{20t}{t_{\max}}} \sum_{j \in K_{\text{best}}} \frac{\text{rand}_j [2 * (h_{ij}(t))^{13} - (h_{ij}(t))^7]}{m_i(t)} \frac{(X_i^d(t) - X_j^d(t))}{\|X_i(t), X_j(t)\|_2} + \beta e^{-\frac{20t}{t_{\max}}} \frac{X_{\text{best}}^d(t) - X_i^d(t)}{m_i(t)} \quad (7)$$

加速度使得原子运动速度及位移发生变化, 这便是 ASO 算法的位置更新的核心过程, 表示为:

$$v_i^d(t+1) = \text{rand}_i^d v_i^d(t) + a_i^d(t) \quad (8)$$

$$X_i^d(t+1) = X_i^d(t) + v_i^d(t+1) \quad (9)$$

3 改进的原子优化算法

3.1 黄金正弦算法

黄金正弦算法 (Golden Sine Algorithm, Golden-SA) 是 Erkan Tanyildizi 于 2017 年提出的新型元启发式算法^[7], 具有求解精度高、收敛速度快等特点。

Golden-SA 算法的核心过程是其解的更新过程, 首先随机产生 s 个个体的位置, 假设优化问题的每个解对应搜索空间中对应个体的位置, 并用 $X_i^d(t)$ 表示 d 维个体空间中第 t 次迭代第 i 个个体的空间位置, 可以表示为 $X_i^d(t) = (X_{i,1}, X_{i,2}, \dots, X_{i,d}) (i = 1, 2, \dots, s; t = 1, 2, \dots, t_{\max})$, $t_{\max}$ 为最大迭代次数, $P^d(t) (t = 1, 2, \dots, t_{\max})$ 为第 t 次迭代全局最优位置, 则第 i 个个体的位置更新公式如下:

$$X_i^d(t+1) = X_i^d(t) * |\sin(R_1)| + R_2 * \sin(R_1) * |x_1 * P_i^d(t) - x_2 * X_i^d(t)| \quad (10)$$

式中, R_1 和 R_2 为随机数, R_1 决定下一次迭代中个体的移动距离, $R_1 \in [0, 2\pi]$, R_2 决定下一次迭代第 i 个个体的位置更新方向, $R_2 \in [0, \pi]$; x_1 和 x_2 是通过引入黄金分割数得到的系数, 这些系数缩小了搜索空间引领个体逐步趋近最优值, 保证了算法的收敛性, 黄金分割数是一个定义为 $(\sqrt{5} - 1)/2$ 的无理数, 记为 τ , $x_1 = a * \tau + b * (1 - \tau)$, a 和 b 初始值设置为 $-\pi$ 和 π , 随后 a 和 b 随着目标值的变化而变化, x_1 和 x_2 也随之更新, 更新方式如下:

Step 1: if current value < target value

then $b = x_2, x_2 = x_1$

$x_1 = a * \tau + b * (1 - \tau)$

else $a = x_1, x_1 = x_2$

式中, $M_i(t) = e^{\frac{\text{Fit}_i(t) - \text{Fit}_{\text{best}}(t)}{\text{Fit}_{\text{worst}}(t) - \text{Fit}_{\text{best}}(t)}}$; $\text{Fit}(t)$ 为原子的适应度函数值。

综上所述, 第 i 个原子在第 t 次迭代时加速度可以表示为:

$$x_2 = a * (1 - \tau) + b * \tau$$

Step2: if $x_1 = x_2$

then $a = \text{rand}_1, b = \text{rand}_2$

$$x_1 = a * \tau + b * (1 - \tau)$$

$$x_2 = a * (1 - \tau) + b * \tau$$

3.2 黄金正弦原子优化算法 (GSASO)

本文将黄金正弦作为局部优化算法融入基本 ASO 中, 即在算法每一次迭代的后期对整个原子个体进行黄金正弦操作。黄金正弦操作的引入可以促进信息在种群中的迅速传播, 使一般原子个体与最优个体进行信息交流, 充分吸收与最优个体位置差信息, 优化算法的寻优方式。基本 ASO 算法前期倾向于全局搜索, 黄金正弦算法的引入可以弥补收敛速度慢的缺陷, ASO 算法后期集中于局部开发通过对个体进行黄金正弦的操作可以增加种群的多样性, 使个体跳出局部最优, 减少算法陷入局部最优的可能性。本文所提出的黄金正弦原子优化算法的具体步骤如下:

步骤 1: 初始化 GSASO 各参数, 如种群规模 s , 最大迭代次数 $t_{\max}$ 等;

步骤 2: 对原子种群进行初始化;

步骤 3: 根据目标函数计算每个个体的适应度函数值, 并保留为当前的最优值及最优解;

步骤 4: 根据公式(7)更新原子运动加速度;

步骤 5: 根据公式(8)更新原子运动速度;

步骤 6: 根据公式(9)更新原子个体位置;

步骤 7: 对步骤 6 所生成的原子个体位置利用 Golden-SA 算法进行黄金正弦操作指引其更新到新的原子个体位置:

$$X_i^d(t+1) = X_i^d(t) * |\sin(R_1)| +$$

$$R_2 * \sin(R_1) * |x_1 * X_{\text{best}}^d(t) - x_2 * X_i^d(t)| \quad (11)$$

步骤 8: 再次计算种群个体的适应度函数值, 根据适应度值的优劣来更新最优解和最优值;

步骤 9: 重复步骤 4-8, 直到达到最大迭代次数

时终止操作；

步骤 10: 输出最优个体位置以及最优适应度值.

GSASO 的算法流程如图 3 所示:

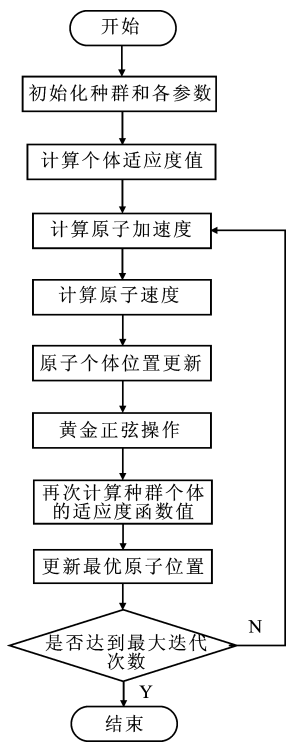


图 3 GSASO 的算法流程图

4 实验仿真及结果分析

4.1 仿真实验环境

本仿真测试环境为:操作系统为 Windows10, CPU 为 Intel(R) Core(TM) i5-8250U, 主频 1.8 GHz, 内存为 4 GB, 仿真软件为 Matlab2015a.

4.2 实验的初始参数设置

本文选取了基本原子优化算法 (ASO)、黄金正弦原子搜索算法 (GSASO)、粒子群算法 (PSO) 进行对比, 为了保证实验的客观性和公平性, 本文将所有算法的种群规模统一设为 50, 迭代次数设为 200, 所有算法共有参数保持一致, 各测试算法的参数设置如表 1 所示:

表 1 各算法实验参数设置

算法	参数
ASO	$\alpha=50, \beta=0.2$
GSASO	$a_0=-\pi, b_0=\pi$
PSO	$\omega=1, \omega_{\text{damp}}=0.99, c_1=1.5, c_2=2$

4.3 测试函数

为了进一步验证 GSASO 的优化效果和稳定性优于其他算法, 本文对多个不同特点的测试函数进行求解函数的对比测试, 测试函数及其具体信息见表 2.

表 2 测试函数及其具体信息

函数	维数	范围	最优值
$f_1=10^6x_i^2+\sum_{i=2}^n x_i^2$	30	$[-100,100]$	0
$f_2=\sum_{i=1}^n ([x_i+0.5])^2$	30	$[-100,100]$	0
$f_3=\sum_{i=1}^n [x_i^2-10\cos(2\pi x_i)+10]$	30	$[-5.12,5.12]$	0
$f_4=\frac{1}{4000}\sum_{i=1}^n x_i^2-\prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}})+1$	30	$[-600,600]$	0
$f_5=-20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^n x_i^2})-\exp(\frac{1}{n}\sum_{i=1}^n \cos(2\pi x_i))+20+e$	30	$[-32,32]$	0
$f_6=-\sum_{i=1}^5 [(X-a_i)(X-a_i)^T+c_i]^{-1}$	4	$[0,10]$	-10.1532

4.4 实验结果与分析

本文对算法的性能在多个测试函数上进行了试验, 为了避免偶然性带来的结果偏差, 算法在每个函数上独立运行 30 次. 表 3 列出了黄金正弦原子优化算法 (GSASO)、原子优化算法 (ASO) 和粒子群算法 (PSO) 这三种算法在多个标准测试函数上经过 30 次独立运行后所得的实验结果.

从寻优最优值、平均值和标准差三个方面对算法进行分析评估, 从实验结果可以看出, 对于 f_2 、 f_3 和 f_4 , GSASO 可以直接搜索到最优值 0, ASO 在求

解 f_2 时最好情况虽然得到了函数最优值, 但其求解平均值和寻优稳定性都劣于 GSASO. 对于 f_1 , ASO 的求解精度最差, 平均精度仅能达到 10^4 这一数量级, 比 PSO 的求解精度还低, 而改进后的 GSASO 算法求解精度可以达到 10^{-24} 这一数量级, 有效弥补了基本 ASO 的缺点. 对于 f_5 , ASO、GASO 和 PSO 都能得到较满意解, 但 GASO 的求解精度和稳定性都明显优于其他两个算法. 对于 f_6 , GASO 的平均求解精度基本接近函数最优值 -10.1532, 表现优秀.

表 3 测试函数实验结果

函数	算法	最优值	平均值	标准差
f_1	ASO	5.740E+03	1.420E+04	5.908E+03
	GSASO	9.925E-26	1.655E-24	1.802E-24
	PSO	9.435E-05	3.241E-03	4.231E-03
f_2	ASO	0	5.000E-01	6.708E-01
	GSASO	0	0	0
	PSO	1.573E-05	2.563E-04	2.331E-04
f_3	ASO	1.990E+01	2.786E+01	5.784E+00
	GSASO	0	0	0
	PSO	2.686E+01	4.368E+01	1.209E+01
f_4	ASO	6.661E-16	1.385E-01	3.117E-01
	GSASO	0	0	0
	PSO	4.427E-07	8.585E-03	7.437E-03
f_5	ASO	1.114E-07	9.241E-02	2.672E-01
	GSASO	5.773E-14	3.597E-13	2.717E-13
	PSO	2.192E-03	9.656E-01	7.364E-01
f_6	ASO	-10.1532	-6.8711	3.447E+00
	GSASO	-10.1532	-10.1523	1.390E-03
	PSO	-0.2647	-0.2647	0

本文提出的黄金正弦原子优化算法(GSASO)与基本原子优化算法和粒子群算法(PSO)相比,其寻优精度和寻优稳定性(标准差)都有明显优势.黄金正弦原子优化算法跳出局部最优的能力最强,具有较好的寻优性能.从图 4 各测试的函数迭代曲线图可以更加直观的看出,本文提出的 GSASO 算法比基本 ASO 算法的收敛精度和收敛速度都要优秀,进一步验证了 GSASO 拥有更好的寻优性能.

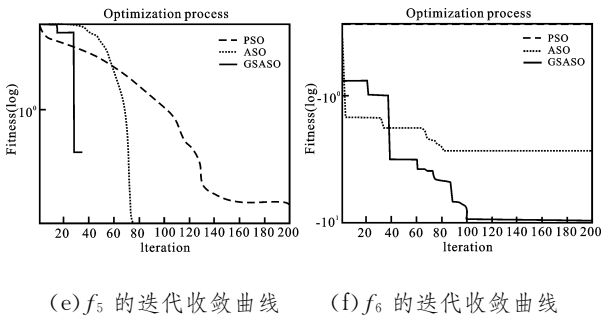
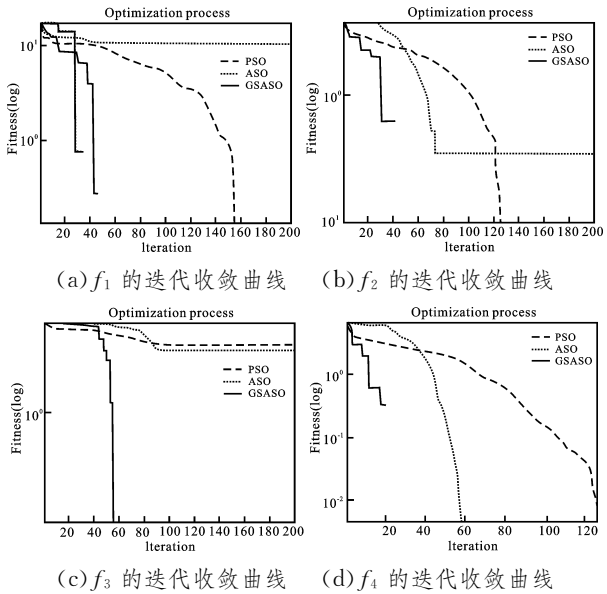


图 4 测试函数的迭代收敛曲线

5 结束语

针对基本原子优化算法易于陷入局部最优和收敛速度慢的问题,本文提出了一种改进的原子优化算法黄金正弦原子优化算法(GSASO),将黄金正弦算法作为一种局部优化算子融入基本 ASO 算法,即在每次迭代后期对每一个原子个体进行黄金正弦操作.黄金正弦操作的引入可以促进信息在种群中的迅速传播,优化算法的寻优方式,使个体跳出局部最优,避免算法早熟,还可以弥补基本算法收敛速度慢的缺陷,提升算法搜索效率.通过对 6 个标准测试函数的仿真实验表明,本文所提出的黄金正弦原子优化算法比基本原子优化算法具有更优秀的寻优性能.

参考文献:

[1] ZHAO W, WANG L, ZHANG Z. A novel atom search optimization for dispersion coefficient estimation in groundwater[J]. Future Generation Computer Systems, 2018, 91(2):601-610.

[2] CICCOTTI G, FERRARIO M, RYCKAERT J P. Molecular dynamics of rigid systems in cartesian coordinates A general formulation[J]. Molecular Physics, 1982, 47(6):1253-1264.

[3] YANG X S. Flower Pollination Algorithm for Global Optimization[C]// International Conference on Unconventional Computation and Natural Computation. Springer-Verlag, 2012:240-249.

[4] PASSINO K M. Biomimicry of bacterial foraging for distributed optimization and control[J]. IEEE Control Systems, 2002, 22(3):52-67.

[5] YANG X S, GANDOMI A H. Bat algorithm: a novel approach for global engineering optimization[J]. Engineering Computations, 2012, 29(5): 464-483.

Memory Write TLP 发送完成包,由主控端 IP 核完成解析并发送至完成队列;随后,设备端控制器发送 MSI-X 中断信号通知主控端 IP 核命令数据传输完成;IP 核收到命令后检查完成队列条目,对成功执行完毕的完成包直接丢弃,未成功的完成包重新进行数据传送,直至完成队列为空,通过 PCIe Memory Write TLP 更新 CQ 头部 DB 寄存器。

4 仿真验证

IP 核收到用户命令后,Busy 信号置高,经 PCIeTx 端口发送更新 DB 寄存器、64 Byte SQ 命令码和用户数据,图 4、图 5 的仿真以 Identify(返回 4 K 数据包)、Write 128 KByte、Read 128 KByte 数据为例;经 PCIeRx 端口接收设备端控制器返回的读取 SQ 头部命令、读用户数据命令、完成包和中断信号。

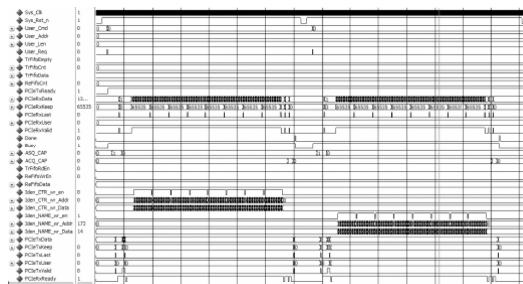


图 4 Identify 命令仿真测试

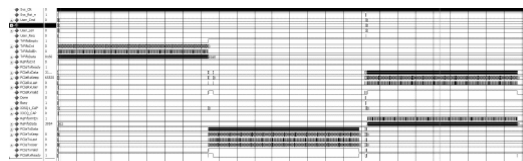


图 5 Write 128KByte、Read 128KByte 命令仿真测试

5 结束语

当前市场下,SATA SSD 已略显疲态,随着市

场存量的不断消解,最终将被 NVMe/PCIe 产品所取代.在航天应用领域,FPGA 以其丰富的逻辑资源与可靠性保证一直被作为主控系统的首选方案.因此,结合 FPGA 的并行加速特性,本文设计了基于 FPGA 的 NVMe 的主控端 IP 核,集成度高、扩展性强、运用灵活方便,特别适合需要大容量存储的嵌入式系统,具有良好嵌入式应用前景.后续该 IP 核还可以进一步拓展应用于 RAID 阵列,设计并实现更高性能的存储技术。

参考文献:

- [1] 董星.基于 FPGA 的存储控制器及相关系统设计技术研究[D].杭州:浙江大学,2016.
- [2] NVMe 标准组织. NVM_Express_Revision_1.3a[EB/OL].(2017-10-24)[2018-08-24] <http://www.nvmexpress.org/about/>.
- [3] SSD FANS.深入浅出 SSD:固态存储核心技术、原理与实战[M].北京:机械工业出版社,2018.
- [4] 王齐.PCIe 体系结构导读[M].北京:机械工业出版社,2010.
- [5] 盘勇军,黄伯铮. VxWorks6.8 操作系统下的 NVMe 驱动设计[J].航空电子技术,2017,48(4):32-33.
- [6] 王磊.基于 VPX 的嵌入式高性能存储技术研究[D].杭州:浙江大学,2018.
- [7] 贾效玲,张刚,张博. SATA 硬盘控制器的 FPGA 实现[J].火力与指挥控制,2016,41(9):182.

作者简介:

崔丹丹 女,(1990-),硕士研究生.研究方向为空间数据存储. E-mail:cuidandan16@csu.ac.cn.

宫永生 男,(1984-),博士,副研究员.研究方向为空间数据存储.

(上接第 25 页)

- [6] GOLDSTEIN H, POOPLE C, SAFKO J, et al. Classical Mechanics, 3rd ed. [J]. American Journal of Physics, 2002, 70(7):782-783.
- [7] TANYILDIZI E, DEMIR G. Golden Sine Algorithm: A Novel Math-Inspired Algorithm[J]. Advances in Electrical & Computer Engineering, 2017, 17(2): 71-78.

作者简介:

肖子雅 女,(1994-),硕士研究生.研究方向为智能算法,项目调度与优化,项目管理.

刘升(通讯作者) 男,(1966-),博士,教授.研究方向为智能计算,机器学习.

E-mail: ml8817568572_1@163.com.