

引用格式: 孔韵雯, 李峭, 熊华钢, 等. 片间综合化互连时间触发通信调度方法[J]. 航空学报, 2018, 39(2): 321590. KONG Y W, LI Q, XIONG H G, et al. Time-triggered communication scheduling method for off-chip integrated interconnection[J]. Acta Aeronautica et Astronautica Sinica, 2018, 39(2): 321590 (in Chinese). doi: 10.7527/S1000-6893.2017.21590

# 片间综合化互连时间触发通信调度方法

孔韵雯<sup>1</sup>, 李峭<sup>1,\*</sup>, 熊华钢<sup>1</sup>, 程子敬<sup>2</sup>

1. 北京航空航天大学 电子信息工程学院, 北京 100083

2. 北京卫星信息工程研究所, 北京 100080

**摘要:** 未来先进的分布式综合化航空电子系统需要实现跨越计算体系结构的信息综合,对微小型智能器件间的综合化互连提出要求。通过构建一种具有开放式接口的芯片间综合化互连结构,提出一种相应的时间触发(TT)通信调度方法。首先建立片间综合化互连模型和时间触发流量传输模型,给出负载均衡的选径方法;随后依据传输路径中各级节点各流量的发送时间偏移量计算流量传输的等待时间可行值,利用遗传算法优化调整各芯片发送端口的调度表相位,更新各流量在发送端口的时间偏移量,缩短最坏情况下的最大等待时间,得到具有全局优化意义的时间触发调度表。与使用Yices等SMT形式化求解器的TT调度表生成方法相比,本方法不会出现长时间不停机而无法判定的问题,且不论对于对称或非对称结构,案例研究表明本方法的可调度规模至少增加30%。此外,以流量传输等待时间占周期的比例作为归一化的传输延迟度量,与既有的基于特征任务的调度方法相比,案例研究表明本方法得到的传输延迟最多仅为后者的2%。

**关键词:** 片间综合化互连; 时间触发通信; 调度方法; 等待时间; 调度表

中图分类号: V247; TP393

文献标识码: A

文章编号: 1000-6893(2018)02-321590-10

综合化航空电子系统经历了分立式、联合式、综合式到高度综合的发展过程<sup>[1-2]</sup>,目前,高度综合的分布式综合模块化航空电子(Distributed Integrated Modular Avionics, DIMA)体系结构,其特征是利用分布式架构将所有综合化模块分布在飞行器中<sup>[3]</sup>,通过时间确定网络实现安全关键性消息的严格周期确定性。文献[4]提到未来航电系统计算体系结构向着智能化、多核处理、微内核和片上系统(System on Chip, SoC)的方向发展。更进一步,随着可综合模块微小型化的发展,例如:美国国防高级研究计划局(DARPA)的MTO办公室推出新型片内微结构的研究项目<sup>[5]</sup>,欧洲e2V公司推出了几百克重的航电多核处理机<sup>[6]</sup>,微小型智能片间互连系统有望成为未来航空航天电子系统分布式综合对象。

迄今为止航电综合化互连技术属于局域网(LAN)或系统域网(SAN)范畴,尚未出现广为认可的跨越处理核内部总线、外部局部总线、I/O总线、SAN和LAN等计算体系结构<sup>[7]</sup>层次的互连技术。针对微小型智能片上系统亟待解决的片间互连问题,串行外设接口(SPI)和Rapid-IO支持的最大速率虽然远高于I<sup>2</sup>C的5 Mbit/s<sup>[8-9]</sup>,但是没有提供固定规则的寻址方案和数据流量控制<sup>[10]</sup>,基于SPI总线和Rapid-IO的消息传输寻址都需要设计人员的特定规划。考虑到未来应用迁移的方便性,本文采用开放性介质无关接口实现芯片间互连,满足跨体系结构层次互连的功能一致性。

对于跨越微小型智能器件的综合化互连系统,文献[11]研究了混合关键性系统分布式实时

收稿日期: 2017-07-07; 退修日期: 2017-10-09; 录用日期: 2017-10-24; 网络出版时间: 2017-10-27 09:37

网络出版地址: <http://hkxb.buaa.edu.cn/CN/html/20180223.html>

基金项目: 国家自然科学基金(91438117,91538202)

\* 通讯作者. E-mail: avionics@buaa.edu.cn

架构,通过时间触发以太网(Time-Triggered Ethernet, TTE)实现了多个多核芯片互连的跨层次体系结构,其侧重于该架构下的可重配置方法。文献[12]研究了片上和片间跨层次体系结构模型,并分别将基于静态调度表和基于优先级的两种调度方法应用于该模型,但前者会引入调度切换开销,后者由于不同层次结构中流量约束优先级不同,可能会对可组合性产生影响。更有甚者,文献[13-14]提出了时间触发片上系统(Time-Triggered System on Chip, TTSoC)架构,通过时间触发片上网络<sup>[15]</sup>(Time-Triggered Network-on-Chip, TTNoC)实现片上多个异构IP块的互连,但与底层硬件联系紧密。

得益于时间触发网络中TT流量调度方法的积累,可以为片间互连网络拓扑中TT流量的调度方法提供参考。如:文献[16-17]采用可满足模理论(Satisfiability Modulo Theories, SMT)求解器,通过网络拓扑和流量配置进行约束,生成满足约束的TT网络时间触发调度表;文献[18]则提出了利用最早期限(EDF)算法调度异步任务的增量化调度方法;文献[19]综合考虑了时间触发控制器处理和通信的约束关系,但片间互连往往是多网段分布式交换结构,无法直接应用上述总线调度结果。对于更细粒度的片间TT流量调度,文献[20-21]基于特征任务的严格周期性任务可调度分析方法可供借鉴。

另外,对于端系统主机组成的TTE网络,虽然TT流量的单跳交换延迟为 $\mu\text{s}$ 量级,但应用层事件触发和时间触发机制配合的等待时间甚至达到流量周期的量级,TTTech公司的操作系统中间件<sup>[22]</sup>可以部分解决上述问题;但对于片上系统组成的网络,应用的操作环境是轻量化的,很难在源、宿节点嵌入纯软件实现的中间件,且具有多级交换转发,为了保证各物理链路上TT流量的严格实时性,只得将等待时间分散于各交换节点。

本文的贡献在于建立基于开放性介质无关接口的芯片间综合化互连模型,并在现有严格周期性任务可调度分析方法<sup>[20-21]</sup>的基础上,提出一种应用于片间互连的TT通信调度方法。本方法采用负载均衡的选径方法,以尽量减小等待时间为目标确定流量的发送时间偏移量,并利用遗传算法优化发送端口的调度表相位,获得具有全局意

义的时间触发调度表。

## 1 片间综合化互连模型

### 1.1 片间互连的拓扑结构

在电路板范围内,芯片之间通过开放式的介质无关接口建立物理链路,进行全双工通信。芯片的物理链路数量和布局根据数据传输的需求确定,在一般情况下是不对称的。取代独立的中心交换机的交换结构也通常不对称,分别就近嵌入于芯片端口内的可编程结构中。选定某一芯片作为片间互连的网关,通过连接板外交换机节点进行跨电路板消息的端到端传输。

将一个芯片间互连结构记为 $G=(V,E)$ , $V=\{v_i\}$ 表示各个芯片, $E=\{e_i\}$ 表示芯片间的双向连接路径。芯片之间的通信是全双工的,相邻芯片间的物理连接可以根据不同的方向标记为不同的通信链路,用 $L=\{l_i\}$ 表示板上通信链路的集合。若 $v_i, v_j \in V$ 为相邻节点,则 $(v_i, v_j) \in L$ 表示 $v_i$ 到 $v_j$ 的通信链路; $(v_j, v_i) \in L$ 表示 $v_j$ 到 $v_i$ 的通信链路。

对于给定芯片 $v_s, v_d \in V$ 间的端到端通信,从 $v_s$ 到 $v_d$ 传输路径可以表示为

$$p = \{(v_s, v_i), \dots, (v_j, v_d)\}$$

式中: $v_s$ 为源节点; $v_d$ 为目的节点;其他芯片为中转节点,角标 $i$ 和 $j$ 为芯片编号。此外,规定流量传输路径不会构成回环。

图1为局部芯片互连拓扑示例,选择 $v_6$ 作为板上网关节点。图中所示为传输路径: $e_1$ 和 $e_2$ 分别为 $v_1, v_2$ 和 $v_1, v_5$ 之间的双向连接通信路径; $v_5$ 为中转节点,包含链路 $l_1=(v_5, v_4)$ 的发送端口。

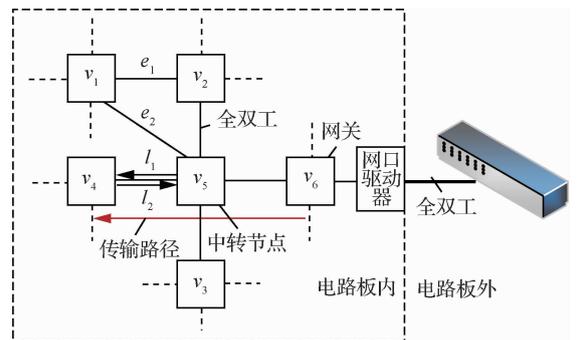


图1 局部芯片互连拓扑

Fig. 1 Topology of local chip interconnection

## 1.2 时间触发流量片间通信模型

假设片间通信的 TT 流量集合为  $F = \{f_1, f_2, \dots, f_n\}$ , 每条流量都由芯片产生或是由网关转发进入板内, 用 4 组元  $f_i = \langle P_i, T_i, c_i, p_i \rangle (1 \leq i \leq n)$  表示,  $P_i$  为流量的优先级, 本文中规定跨板流量具有最高优先级, 周期具有次高优先级, 周期越小优先级越高;  $T_i$  和  $c_i$  分别为周期和执行时间;  $p_i$  为传输路径, 包含了  $q (q \geq 1)$  段通信链路, 源节点为  $v_{s,i}$ , 目的节点为  $v_{d,i}$ , 经过的跳数为  $h = q + 1$ 。流量传输路径可以由设计根据需求静态离线确定, 对于没有确定路径的流量, 将在 2.1 节中说明根据负载均衡原则确定路径的方法。

芯片之间需要进行时钟同步, 专用同步线是可选的近距离时钟同步的简易方法; 然而, 考虑到抗摧毁能力和未来应用迁移所需的开放性, 利用片上的可编程资源实现压缩主控器 (Compress Master, CM)、同步主控器 (Synchronization Master, SM) 和同步客户端 (Synchronization Client, SC), 通过协议控制帧 (Protocol Control Frame, PCF) 实现各个芯片间的时钟同步, 形成符合 SAE AS6802 标准<sup>[23]</sup> 的同步过程, 使之不仅在本板电路板上实现同步, 而且可以与满足该标准的时间触发网络实现同步。目前, 对于 100 Mbps 码速率的 TTE, 分布式同步精度为  $1 \mu\text{s}$ 。此外, TT 流量通信过程有以下规则:

- 1) 各个芯片端口发送的 TT 流量均满足严格周期性, 即同一个 TT 流量的两个连续消息之间的时间间隔是固定的, 且等于周期。
- 2) 本调度方法中规定选定作为网关的片上系统不作为消息传输路径中的中转节点中继转发消息。
- 3) 跨板流量具有最高的优先级, 在所有等待调度的 TT 流量中优先被安排。

## 1.3 严格周期任务的可调度性

如果将时间触发流量的传输服务作为任务, 传输延迟作为任务的“执行”时间, 则也可借鉴多处理器任务集合可调度性的判定方法。在下面的讨论中, 令“任务”与 TT 流量的含义相同, 而消息作为流量的实例, 相当于任务的“作业”。

假设  $T = \{\tau_1, \tau_2, \dots, \tau_n\}$  是系统中已经存在

的任务集合, 其中  $\tau_i = \langle T_i, c_i, t_i \rangle (1 \leq i \leq n)$ ,  $t_i$  为开始时间偏移量。满足关系  $0 < c_i \leq T_i$  和  $0 \leq t_i \leq T_i - c_i$ 。 $\tau'_r = \langle 1, T_r \rangle$  为需要进行可调度性判定的任务  $\tau_r = \langle c_r, T_r \rangle$  的特征任务<sup>[20]</sup>,  $r$  为判定任务的编号。

此外, 用函数  $\text{LLC}(S)$  表示一个非空集合  $S$  中连续整数的最长长度, 函数  $\text{SLC}(S, l)$  表示一个非空集合  $S$  中长度大于  $l$  的连续整数中的最小整数<sup>[20]</sup>, 函数  $\text{LC}(S)$  表示一个非空集合  $S$  中各段连续整数的长度。

基于特征任务的可调度性分析中的关键参数分别为任务  $\tau'_r$  因任务  $\tau_i$  占用而不能使用的时间  $\text{ITU}(i, r)$  和所有使  $\tau'_r$  可调度的空余时间  $Q_T(r)$ , 过程主要包含如下步骤:

- 1) 利用原任务的特征任务, 采用取模运算求出  $T_r$  上被任务  $\tau_i$  占用的时间  $\text{ITU}(i, r)$ 。
- 2) 求出特征任务在可调度时的所有空余时间。
- 3) 计算特征任务可用空余时间的最大长度, 判断任务的可调度性。

**定理 1**<sup>[20-21]</sup> 任务  $\tau_r = \langle c_r, T_r \rangle$  与集合  $T$  中所有任务是可调度的, 当且仅当:

$$c_r \leq \text{LLC}(Q_T(r)) \quad (1)$$

式中:  $\text{LLC}(Q_T(r))$  为空余时间  $Q_T(r)$  的最长长度。

## 2 TT 流量调度方法

片间互连的 TT 流量调度方法在 1.3 节严格周期任务可调度性判定方法的基础上, 调度得到由分布于各端口的循环调度表组成的全局时间触发调度表。该方法主要有以下 3 个步骤:

- 1) 利用基于特征任务的可调度性分析, 计算端口负载率, 根据负载均衡原则确定流量传输路径。
- 2) 以等待时间局部最优为目标, 依据优先级排序以增量化方式初步确定流量发送时间偏移量, 获得各端口的循环调度表。
- 3) 利用遗传算法调整各端口调度表相位, 更新流量在各端口发送时间偏移量, 得到全局优化意义的时间触发调度表。

### 2.1 流量传输路径的选择

对于 TT 流量跳数相同的可能端到端路径, 以负载均衡为目标选择流量传输路径。为了提高

路径选择算法的效率,采用广度优先,在不产生回环路径的前提下,逐级搜索可能经过的通信链路,判定 TT 流量在通信链路发送端口的可调度性。若存在无法调度的端口,则通过“剪枝”,剔除包含该端口所在传输路径。通过累加计算可行端到端路径的流量负载率,选择负载率最小的路径为流量的传输路径。

### 2.1.1 TT 流量可调度性判定算法

参考 1.3 节严格周期任务的可调度性判定多次和定理 1<sup>[20-21]</sup>,算法 1 用于判定 TT 流量在通信链路  $l_i$  发送端口的可调度性,其计算的时间复杂度为  $O(NT_{\max})$ ,  $T_{\max}$  为流量集合里的最大周期。

假定  $l_i$  发送端口已存在的 TT 流量集合为  $F_{l_i} = \{f_{l_i}^1, f_{l_i}^2, \dots, f_{l_i}^n\}$ , 待判定的流量为  $f_r = \langle P_r, T_r, c_r, p_r \rangle$ 。经过计算后,可得到的空余时间集合  $Q_{F_{l_i}}(r)$  和发送时间偏移量可取值构成的集合  $O_{F_{l_i}}(r)$ 。算法 1 的伪代码如下。

算法 1 TT 流量可调度性判定算法

输入: TT 流量集合  $F_{l_i} = \{f_{l_i}^1, f_{l_i}^2, \dots, f_{l_i}^n\}$  和待判定流量  $f_r = \langle P_r, T_r, c_r, p_r \rangle$ 。  
 输出: 流量  $f_r$  是否能被调度,若能调度则输出集合  $Q_{F_{l_i}}(r)$  和  $O_{F_{l_i}}(r)$ 。

- 1  $b_r \leftarrow 0, Q_{F_{l_i}}(r) \leftarrow [0, T_r - 1], O_{F_{l_i}}(r) \leftarrow \emptyset;$
- 2 **for**  $j=1$  to  $n$ , **do**
- 3  $E_j \leftarrow \cup e_j^k \quad k \in [0, \frac{\text{LCM}(T_r, T_j)}{T_j} - 1];$
- 4  $\text{ITU}(j, r) \leftarrow \{x \mid x = (e_j^k) \bmod (T_r), \forall e_j^k \in E_j\};$
- 5 **end**
- 6  $Q_{F_{l_i}}(r) \leftarrow Q_{F_{l_i}}(r) \setminus \text{ITU}(j, r);$
- 7 **if**  $c_r \leq \text{LLC}(Q_{F_{l_i}}(r))$  **then**
- 8  $b_r \leftarrow 1;$
- 9  $O_{F_{l_i}}(r) \leftarrow \text{SLC}(Q_{F_{l_i}}(r), c_r);$
- 10 **end**
- 11 **return**  $b_r, Q_{F_{l_i}}(r)$  and  $O_{F_{l_i}}(r);$

其中:  $E_j$  为流量  $f_j$  在周期  $T_j$  和  $T_r$  的最小公倍数内所有消息占用的时间;  $e_j^k$  为流量  $f_j$  的第  $k$  条消息所占用的时间,表达式分别为

$$E_j = \cup e_j^k \quad k \in [0, \frac{\text{LCM}(T_r, T_j)}{T_j} - 1] \quad (2)$$

$$e_j^k \in [t_j + kT_j, t_j + kT_j + c_j) \quad (3)$$

### 2.1.2 TT 流量通信路径选择算法

采用广度优先算法,搜索得到待安排路径流量  $f_r = \langle P_r, T_r, c_r, p_r \rangle$  可能的  $k$  条端到端传输路径集合  $R_r = \{p_1, p_2, \dots, p_k\}$ , 以及所有可能路径中包含的  $q$  段物理链路  $l_i (1 \leq i \leq q)$ 。

算法 2 用于选择 TT 流量的通信路径,在算法 1 的基础上先判定链路的可调度性,在剔除包含不可调度的链路的路径后,选择负载率最小的可行端到端路径为流量的传输路径,其计算的时间复杂度为  $O(N^2)$ 。算法 2 的伪代码如下。

算法 2 TT 流量通信路径选择算法

输入: TT 流量集合  $F_{l_i} = \{f_{l_i}^1, f_{l_i}^2, \dots, f_{l_i}^n\}$ , 待安排流量  $f_r = \langle P_r, T_r, c_r, p_r \rangle$  和路径集合  $R_r = \{p_1, p_2, \dots, p_k\}$ 。  
 输出: 流量  $f_r$  选择的通信路径  $p_r$ 。

- 1  $p_r \leftarrow [], U_{l_i} \leftarrow 0, U_{p_j} \leftarrow 0, U_{p_r} = A, k_1 = k;$
- 2 **for**  $i=1$  to  $q$  **do**
- 3 调用算法 1 判定  $f_r$  在链路  $l_i$  的可调度性;
- 4 **if**  $b_r = 1$  **then**
- 5  $U_{l_i} \leftarrow 1 - \frac{\sum L_{l_i}^m}{T};$
- 6 **end**
- 7 **else**
- 8  $U_{l_i} \leftarrow A;$
- 9 **end**
- 10 **for**  $j=1$  to  $k_1$  **do**
- 11 **if**  $l_i \in p_j$  **then**
- 12 **if**  $U_{l_i} = A$  **then**
- 13  $\text{delete } p_j;$
- 14  $k_1 \leftarrow k_1 - 1;$
- 15 **end**
- 16 **else**
- 17  $U_{p_j} \leftarrow U_{p_j} + U_{l_i};$
- 18 **end**
- 19 **end**
- 20 **end**
- 21 **end**
- 22 **for**  $j=1$  to  $k_1$  **do**
- 23 **if**  $U_{p_j} < U_{p_r}$  **then**
- 24  $U_{p_r} \leftarrow U_{p_j}, p_r \leftarrow p_j;$
- 25 **end**
- 26 **end**

其中:  $A$  为预先根据可能负载率设定的较大常数;  $U_{l_i}$  为链路  $l_i$  发送端口的负载率, 即

$$U_{l_i} = 1 - \frac{\sum_{k_i} L_{k_i}^{l_i}}{T} \quad (4)$$

式中:  $L_{k_i}^{l_i} \in LC(Q_{F_{l_i}}(f_{k_i}^{l_i}))$  为算法 1 得到的各段空余时间的长度。

## 2.2 发送时间偏移量的确定

### 2.2.1 发送时间偏移量的初步确定

以流量传输的等待时间局部最优为目标初步确定流量在各节点的发送时间偏移量, 记流量  $f_r$  在中转节点  $v_i$  的发送时间偏移量为  $t_i^{f_r}$ , 并为进一步优化等待时间准备数据。

在片间的传输过程中, 对流量传输的端到端延迟影响最大的是流量在中转节点等待发送的缓存时间, 记流量  $f_r$  在传输路径  $p_r$  的等待时间为  $wt_r(p_r)$ , 流量  $f_r$  在中转节点  $v_i (l_i = (v_i, v_j))$  等待时间为  $wt_r(l_i)$ , 表示流量  $f_r$  在到达  $v_i$  后等待发送的时间。对于包含  $q$  段通信链路的路径  $p_r$ ,  $wt_r(p_r)$  由  $wt_r(l_i)$  累加得到, 即

$$wt_r(p_r) = wt_r(l_2) + wt_r(l_3) + \dots + wt_r(l_q) \quad (5)$$

对于静态调度的流量, 在发送节点端口的发送时间偏移量为

$$t_i^{f_r} = \min(O_{F_{l_i}}(r)) \quad (6)$$

在中转节点端口的发送时间偏移量为

$$t_i^{f_r} = \min(o \in O_{F_{l_i}}(r) \mid o \geq t_{i-1}^{f_r} + c_r) \quad (7)$$

式中:  $t_{i-1}^{f_r}$  为流量  $f_r$  在上一级节点的发送时间偏移量。

以例 1 说明流量逐级初步确定发送时间偏移量的过程。待调度流量  $f_r = \langle 2, 12, 2, p_r \rangle$ , 传输路径  $p_r = \{(v_{s,1}, v_2), (v_2, v_{d,3})\}$  如图 2 所示。源节点  $v_{s,1}$  发送端口中已存在的流量集合为  $F_{l_1} = \{f_1^1, f_2^1\}$ , 中转节点  $v_2$  发送端口中已存在的流量集合为  $F_{l_2} = \{f_1^2, f_2^2\}$ 。其中  $f_1^1 = \langle 1, 6, 1, p_1^1 \rangle$ , 在  $v_{s,1}$  的发送时间偏移量  $t_1^1 = 0$ ;  $f_2^1 = \langle 1, 4, 1, p_2^1 \rangle$ ,  $f_2^2 = \langle 1, 8, 1, p_2^2 \rangle$ , 在  $v_2$  的发送时间偏移量分别为  $t_1^2 = 0$  和  $t_2^2 = 1$ 。

图 3 给出了例 1 根据算法 1 计算得到的结果, 其中图 3(a) 给出了  $v_{s,1}$  发送端口可用的空余

时间; 图 3(b) 为中转节点  $v_2$  发送端口可用的空余时间。

经过计算, 例 1 流量  $f_r$  在源节点发送端口(连接物理链路  $l_1$ ) 发送时间偏移量为  $t_1^{f_r} = \min(O_{F_{l_1}}(r)) = 1$ ; 忽略足够小的物理链路传输延迟, 中转节点发送端口(连接物理链路  $l_2$ ) 的发送时间偏移量为  $t_2^{f_r} = \min(o \in O_{F_{l_2}}(r) \mid o \geq t_1^{f_r} + c_r) = 5$ , 其中, 流量的执行时间  $c_r = 2$ 。

由此得到,  $f_r$  中的消息在路径  $p_r$  的最小可行等待时间  $wt_r(p_r) = wt_r(l_2) = 2$ 。

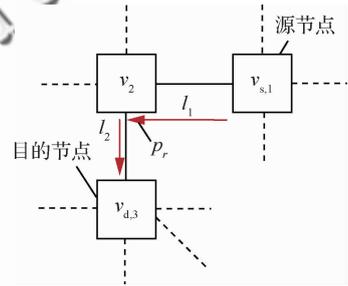


图 2 消息传输路径

Fig. 2 Message transfer path

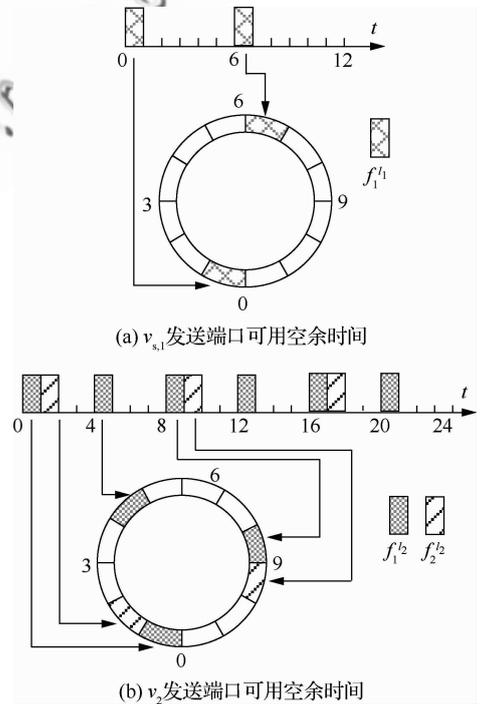


图 3 消息等待时间计算过程

Fig. 3 Calculation process for message waiting time

## 2.2.2 等待时间的优化

发送端口的循环调度表决定了该输出物理链路上 TT 流量的严格周期性,而且各个循环调度表之间的相对相位影响了消息的等待时间。为了缩短最坏情况下的等待时间,采用遗传算法<sup>[24-25]</sup>调整芯片各发送端口的调度表相位,得到各端口近似最优的一组相位值。

给定相位值后,各流量在发送端口的偏移量依次循环移位更新,得到全局优化意义的时间触发调度表,此时最坏情况下的最大等待时间最短。

设遗传算法中的个体为各个端口调度表的相位,记为  $\text{phase}_{v_r}^i$ ; 适应度函数来源于片间综合化互连结构中所有消息中最大等待时间约束,有向控制调节下一代个体的生成。

当一代相位确定后,代入式(8)进行各端口流量  $f_r = \langle P_r, T_r, c_r, p_r \rangle$  的发送时间偏移量的更新

$$t_{r,\text{new}}^i = \min\left(\frac{t_r^i + \text{phase}_{v_r}^i + j \times T_r}{T_{n_i}}\right) \quad (8)$$

式中:  $t_{r,\text{new}}^i$  为流量  $f_r$  更新后的发送时间偏移量;  $T_{n_i}$  为该端口调度安排的第  $n$  条流量的周期,且该端口调度的流量集合为  $F_{v_i} = \{f_1^i, f_2^i, \dots, f_{n_i}^i\}$ 。

偏移量更新后,采用式(8)计算等待时间,进而得到其中的最大等待时间  $\text{wt}_{\max}$ 。从优化目标可知,  $\text{wt}_{\max}$  越大,遗传算法中基因保留在子代的可能性越小,因此设置适应度函数为

$$\text{FitFunc} = \begin{cases} C_{\max} - \text{wt}_{\max} & \text{wt}_{\max} < C_{\max} \\ 0 & \text{wt}_{\max} \geq C_{\max} \end{cases} \quad (9)$$

式中:  $C_{\max}$  为预先根据可能结果设定的较大常数。

根据父代个体所得的适应度值降序的排列,利用选择算子和交叉算子生成子代,经过多代遗传后适应度函数收敛,从而得到近似的最优值。

## 3 案例研究

### 3.1 片间综合化互连模型仿真

设基于芯片间消息实际传输速率为 100 Mbit/s,以太网帧的帧长范围为 64~1 518 字节,则每一帧传输的执行时间范围为 5.12~121.44  $\mu\text{s}$ , TT 流量的周期通常为 1~128 ms。考虑到仿真主要是为了验证该调度方法的有效

性,需要在逼真度和计算复杂度之间取得权衡,因此将消息的执行时间和周期以  $\mu\text{s}$  为单位进行上取整为  $c_r \in [6, 122]$ , 且  $c_r \in \mathbf{N}$ ; 在工程中,取整后增加的冗余量可作为消息的保护间隔。为了在仿真中使各流量的周期存在一定的公约数关系且具有多样性,使  $T_i = 2^n \times 3^m \times 1\,000$ , 其中  $0 \leq n \leq 7, 0 \leq m \leq 2$ , 且  $T_i \in [1\,000, 128\,000]$ 。

仿真规模对应于  $3 \times 3$  网格的实验原型,该原型可配置为对称互连结构及非对称互连结构,分别如图 4(a)和图 4(b)所示。

为了进行对比,其对照组利用 SMT 工具求解各端口调度表<sup>[16-18]</sup>,值得说明的是,现有 SMT 工具仅面向端系统主机之间的互连,计算中将“端系统”等效为芯片的发送端口,“交换机”等效于芯片内部的交换结构,但规定连接于同一“交换机”的“端系统”间不产生外部通信流量。

类似于文献[16],SMT 的单跳延迟下界设置为 122  $\mu\text{s}$ (最长帧执行时间),不可判定情况下的停机时间为 1.5 h,但为了允许中转节点缓冲等待,将单跳延迟上界设为最长帧的 4 倍(文献[16]中的最大选项)。

针对对称与非对称的互连结构,调度规模如表 1 所示,分别采用 SMT 和本文提出的调度方法得到各组流量中的最大等待时间。其中,对称和非对称结构中的最大等待时间分别记为  $\text{wt}_{\max,\text{S}}$  和  $\text{wt}_{\max,\text{A}}$ , 以  $\mu\text{s}$  为单位,“-”表示 SMT 不可判定超时。

由仿真结果可以看出,在两种结构中,当规模较小时,两种方法调度得到的等待时间相近;而流量规模较大时,利用 SMT 进行调度的传输等待时间较小于本文调度方法得到的等待时间,但本文调度方法的可调度流量规模相较于 SMT 至少增加 30%。

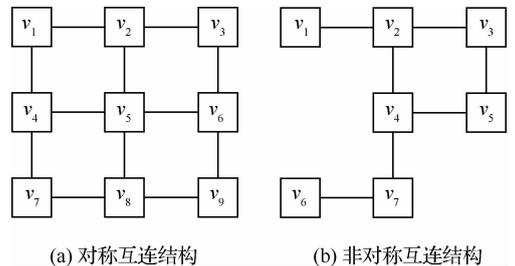


图 4 对称和非对称互连拓扑结构

Fig. 4 Topology structure of symmetrical and asymmetrical interconnection

表 1 可调度流量规模

Table 1 Scheduled message size

流量规模	SMT		本文方法	
	$w_{t_{max},S}/\mu S$	$w_{t_{max},A}/\mu S$	$w_{t_{max},S}/\mu S$	$w_{t_{max},A}/\mu S$
100	1 085	1 333	784	894
200	1 867	2 039	601	2 248
300	1 659	2 437	1 522	3 879
400	1 856	--	2 360	5 973
500	2 133	--	3 733	9 629
600	2 273	--	3 638	不可调度
700	--	--	3 800	不可调度
800	--	--	5 294	不可调度

此外,与 SMT 相比,本调度方法可以明确得到不可调度的结论,且导致不可调度的流量明确,便于根据实际调度情况及时调整。

对照组采用 Yices SMT 求解器,核心算法为优化后的同余闭包算法<sup>[26]</sup>,其时间复杂度为  $O(N^3 \lg N)$ ,然而,当它具体应用于 TT 调度问题时,会出现非线性的离散数学约束,即使仅搜索可行解,也被认为是 NP 问题<sup>[18]</sup>。对于本文提出的调度方法,依次执行算法 1 和算法 2 即可得到可行解,整体时间复杂度为  $O(N^2)$ 。在对称与非对称拓扑结构中,SMT 与本文提出的调度方法对流量进行调度规划的运行时间如图 5 所示,本方法对流量进行调度规划的时间远小于 SMT,并且在对称与非对称结构中,流量规模的增加并不会导致调度规划的时间明显增长。

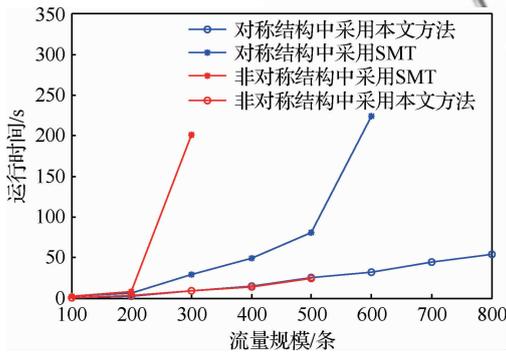


图 5 运行时间对比

Fig. 5 Comparison of execution time

### 3.2 等待时间仿真结果对比

利用 20 组 600 条 TT 流量,对比本文提出的

逐跳计算流量的发送时间偏移量方法(方法 1)和文献[20]中既有的基于特征任务的调度方法(方法 2)对等待时间的影响。

以流量传输等待时间占周期的比例作为归一化的传输延迟度量,图 6 为分别为采用两种方法调度 2~4 跳流量的平均传输延迟;其中 10 组仿真结果中,每组流量中的最大等待时间  $w_{t_{max}}$  及对应的归一化传输延迟见表 2。

在未经过遗传算法优化前方法 1 得到的传输延迟不超过 0.01,流量等待时间没有随着传输跳数的增加而明显增加,而方法 2 得到的传输延迟至少为 1.99,传输路径每增加 1 跳,等待时间显著增加。本方法与既有的基于特征任务的调度方法相比,传输延迟缩短到后者的 2% 以下,得到每条流量的局部最优等待时间。

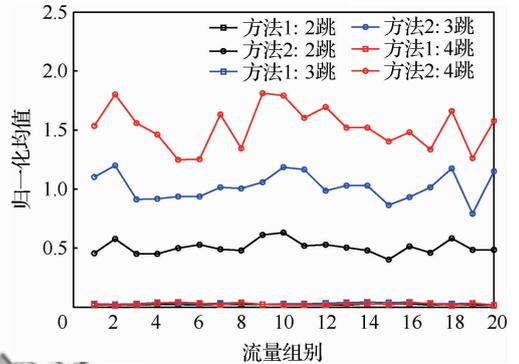


图 6 等待时间归一化均值

Fig. 6 Normalized average of message waiting time

表 2 最大等待时间

Table 2 Maximum waiting time

流量组别	方法 1			方法 2		
	$w_{t_{max}}/\mu S$	$\frac{w_{t_{max}}}{T_i}$	跳数	$w_{t_{max}}/\mu S$	$\frac{w_{t_{max}}}{T_i}$	跳数
1	2 611	0.019	4	286 873	2.99	4
2	4 449	0.035	3	286 855	2.99	4
3	2 894	0.054	2	286 990	2.99	4
4	3 142	0.033	3	255 943	2.00	4
5	2 765	0.022	3	191 705	2.00	4
6	2 669	0.021	2	255 142	1.99	3
7	3 640	0.037	3	255 918	2.00	4
8	3 197	0.033	3	287 218	2.99	4
9	2 932	0.020	3	256 911	2.00	4
10	3 033	0.024	3	255 844	2.00	4

### 3.3 遗传算法优化实验

针对 3.1 节非对称结构中调度的一组 500 条 TT 流量,利用方法 1 调度得到的最大等待时间为  $11\ 173\ \mu\text{s}$ ,利用方法 2 调度得到的最大等待时间为  $25\ 8133\ \mu\text{s}$ 。采用遗传算法优化两种方法调度得到的最大等待时间,其优化结果如图 7 所示。

方法 1 得到的最大等待时间经过遗传算法优化后为  $9\ 645\ \mu\text{s}$ ,优化了 13.7%,方法 2 得到的最大等待时间经过 1 672 代的遗传迭代优化后,尽管该方法调度流量中的最大等待时间在优化后减小了 35.8%,仍然远大于方法 1 得到的最大等待时间,进一步验证了本文方法能够缩短流量传输等待时间这一优势。

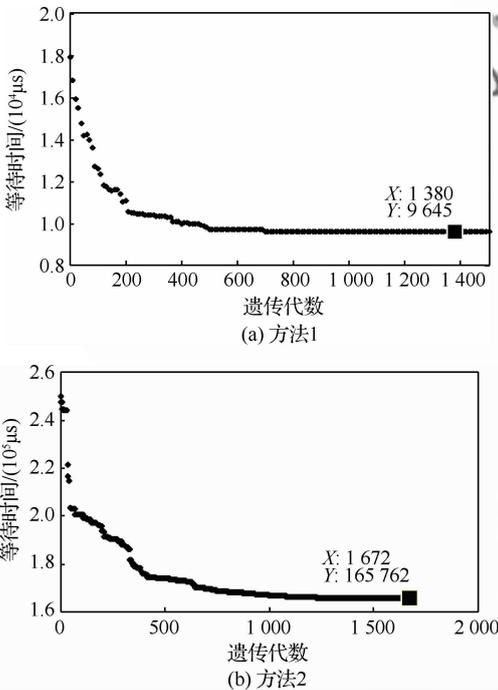


图 7 方法 1 和方法 2 最大等待时间优化过程

Fig. 7 Optimization process of maximum waiting time for Methods 1 and 2

## 4 结论

1) 本文建立了可以用于片间、片上系统和模块联合工作的片间综合化互连结构模型,提出了适用于该结构的一种 TT 通信调度方法,其将缩短等待时间作为优化目标,规划流量传输路径,选择并调整流量发送时间偏移量,获得具有全局最

优意义的时间触发调度表,并通过了案例仿真进行验证。

2) 本方法能够在有限时间内得到是否可调度的判定结论;通过案例研究,对于可调度的情况,本方法的可调度流量规模比 SMT 方法至少增加了 30%;即使不可调度,本方法能够明确指出导致不可调度的流量,便于快速设计调整。

3) 案例研究结果表明,以流量传输等待时间占周期的比例作为归一化的传输延迟度量,本方法得到的传输延迟小于 0.01,与既有的基于特征任务的调度方法相比,缩短到后者传输延迟的 2% 以下。

## 参 考 文 献

- [1] 熊华钢,王中华. 先进航空电子综合技术[M]. 北京:国防工业出版社,2009:2-13.  
XIONG H G, WANG Z H. Advanced avionics integration techniques[M]. Beijing: National Defense Industry Press, 2009: 2-13 (in Chinese)
- [2] 王国庆,谷青范,王森,等. 新一代综合化航空电子系统构架技术研究[J]. 航空学报,2014,35(6):1473-1486.  
WANG G Q, GU Q F, WANG M, et al. Research on the architecture technology for new generation integrated avionics system[J]. Acta Aeronautica et Astronautica Sinica, 2014, 35(6): 1473-1486 (in Chinese).
- [3] WOLFG R, JAKOVLJEVIC M. Distributed IMA and DO-297: Architectural, communication and certification attributes[C]//Proceedings Digital Avionics Systems Conference. Piscataway, NJ: IEEE Press, 2008:1. E. 4-1-1. E. 4-10.
- [4] 蒲小勃. 现代航空电子系统与综合[M]. 北京:航空工业出版社,2013:70-86.  
PU X B. Modern avionics system and integration [M]. Beijing: Aviation Industry Press, 2013: 70-86 (in Chinese).
- [5] DARPA M T O. Another big shrink: Tiling chiplets into next-generation microsystems [EB/OL]. (2016-07-19) [2017-07-05]. <http://www.darpa.mil/news-events/2016-07-19>.
- [6] E2V. E2V and Adeneo partner to create the world's smallest, multicore computer for aerospace applications[EB/OL]. (2016-07-18) [2017-07-05]. <https://www.e2v.com/news/e2v-and-adeneo-partner-to-create-the-worlds-smallest-multicore-computer-for-aerospace-applications/>.
- [7] 林闯,贾子晓,孟坤. 自适应的未来网络体系架构[J]. 计算机学报,2012,35(6):1077-1093.  
LIN C, JIA Z X, MENG K. Research on adaptive future

- internet architecture[J]. Chinese Journal of Computer, 2012, 35(6): 1077-1093 (in Chinese).
- [8] HIERGEIST S, HOLZAPFEL F. Fault-tolerant FCC Architecture for future UAV systems based on COTS SoC [C]//Proceedings Architecture of Computing Systems. Berlin: Springer International Publishing, 2016: 1-5.
- [9] SCHECKEL T. Serial RapidIO: Benefiting system interconnects [C]//Proceedings SOC Conference. Piscataway, NJ: IEEE Press, 2005: 317-318.
- [10] LEENS F. An introduction to I 2 C and SPI protocols[J]. IEEE Instrumentation & Measurement Magazine, 2009, 12(1): 8-13.
- [11] DURRIEU G, FOHLER G, GALA G, et al. DREAMS about reconfiguration and adaptation in avionics[C]//Proceedings Embedded Real Time Software and Systems. Toulouse: European Congress Press. 2016: 48-57.
- [12] OLIVER R S, CRACIUNAS S S. Hierarchical scheduling over off- and on-chip deterministic networks[J]. Acm Sigbed Review, 2016, 13(4): 14-19.
- [13] OBERMAISSER R, EL SALLOUM C, HUBER B, et al. The time-triggered system-on-a-chip architecture[C]//Proceedings IEEE International Symposium on Industrial Electronics. Piscataway, NJ: IEEE Press, 2008: 1941-1947.
- [14] WASICEK A, EI-SALLOUM C, KOPETZ H. A system-on-a-chip platform for mixed-criticality applications[C]//Proceedings IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing. Piscataway, NJ: IEEE Press, 2010: 210-216.
- [15] SCHOEBERL M. A time-triggered network-on-chip[C]//Proceedings International Conference on Field Programmable Logic and Applications. Piscataway, NJ: IEEE Press, 2007: 377-382.
- [16] STEINER W. An evaluation of SMT-based schedule synthesis for time-triggered multi-hop networks[C]//Proceedings Real-Time Systems Symposium. Piscataway, NJ: IEEE Press, 2011: 375-384.
- [17] POZO PÉREZ F M, RODRIGUEZ NAVAS G, HANSSON H, et al. Schedule synthesis for next generation time-triggered networks[C]//IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA). Piscataway, NJ: IEEE Press, 2015: 1-8.
- [18] CRACIUNAS S S, OLIVER R S. SMT-based task- and network-level static schedule generation for time-triggered networked systems[C]//Proceedings International Conference on Real-Time Networks and Systems. Versailles: RTNS Press, 2014: 45-54.
- [19] HU M, LUO J, WANG Y, et al. Scheduling periodic task graphs for safety-critical time-triggered avionic systems[J]. IEEE Transactions on Aerospace & Electronic Systems, 2015, 51(3): 2294-2304.
- [20] CHEN J, DU C, XIE F, et al. Schedulability analysis of non-preemptive strictly periodic tasks in multi-core real-time systems[J]. Real-Time Systems, 2016, 52(3): 239-271.
- [21] 陈进朝, 杜承烈. 单处理器平台下的严格周期任务可调度性判定[J]. 计算机工程, 2016, 42(5): 288-291.
- CHEN J C, DU C L. Schedulability test for strictly periodic tasks in uniprocessor systems [J]. Computer Engineering, 2016, 42(5): 288-291 (in Chinese).
- [22] TTE-COM. TTE-COM A653 for VxWorks 653 2.4 [EB/OL]. (2016-10-25) [2017-07-05]. Vienna, Austria: TTTech Computertechnik AG, 2016. <https://www.tttech.com/products/aerospace/development-test-vv/middleware/tte-com-a653-for-vxworks-653/>.
- [23] SAE. Time-triggered Ethernet: AS6802[S]. Warrendale, PA: SAE International, 2011.
- [24] DING S, YIN X, XU H, et al. A hybrid GA-based scheduling method for static segment in FlexRay systems [C]//Proceedings Control and Decision Conference. Piscataway, NJ: IEEE Press, 2010: 1548-1552.
- [25] 代真, 何锋, 张宇静, 等. AFDX 虚拟链路路径实时寻优算法[J]. 航空学报, 2015, 36(6): 1924-1932.
- DAI Z, HE F, ZHANG Y J, et al. Real-time path optimization algorithm of AFDX virtual link[J]. Acta Aeronautica et Astronautica Sinica, 2015, 36(6): 1924-1932 (in Chinese).
- [26] DUTERTRE B. Yices 2.2[C]//Proceedings International Conference on Computer Aided Verification. Berlin: Springer International Publishing, 2014: 737-744.

(责任编辑: 苏磊)

# Time-triggered communication scheduling method for off-chip integrated interconnection

KONG Yunwen<sup>1</sup>, LI Qiao<sup>1,\*</sup>, XIONG Huagang<sup>1</sup>, CHENG Zijing<sup>2</sup>

1. School of Electronics and Information Engineering, Beihang University, Beijing 100083, China

2. Beijing Institute of Satellite Information Engineering, Beijing 100080, China

**Abstract:** The advanced distributed integrated avionics system needs to achieve information integration across computing architectures. Integrated interconnection among micro-smart devices is thus required. By constructing an off-chip integrated interconnection structure with the open interface, a corresponding time-triggered message scheduling method is proposed. First, the off-chip interconnection model and the Time-Triggered (TT) communication model are established, and the load balancing path selection method is given. Then, the feasible value of the waiting time is calculated according to the flow's transmission offset of chips in the transmission path. The genetic algorithm is used to adjust the phase of scheduling tables of chips' send ports, whereby the flow's transmission offset can be updated and the maximum waiting time of the worst case can be shortened. The time-triggered scheduling table with global optimization significance can be obtained. Compared with the methods of generating TT scheduling tables using the SMT formalizer, such as Yices, this method does not work for a long time without stopping and can not cause undefined problems. The case study shows that the scheduler scale of this method increases by at least 30% for both symmetric and asymmetric structures. In addition, the ratio of the waiting time to the time period is taken as the normalized transmission delay metric. The case study also shows that the propagation delay value deduced by this method reduces to less than 2% of the one deduced by existing task-based scheduling methods.

**Keywords:** off-chip integrated interconnection; time-triggered communication; scheduling method; waiting time; scheduling table

**Received:** 2017-07-07; **Revised:** 2017-10-09; **Accepted:** 2017-10-24; **Published online:** 2017-10-27 09:37

**URL:** <http://hkxb.buaa.edu.cn/CN/html/20180223.html>

**Foundation items:** National Natural Science Foundation of China (91438117, 91538202)

\* **Corresponding author.** E-mail: [avionics@buaa.edu.cn](mailto:avionics@buaa.edu.cn)