

FHT 在 WCDMA 辅同步中的应用及 FPGA 实现

谷涛

北京邮电大学信息工程学院, 北京 (100876)

E-mail: gutao1984@gmail.com

摘要: 快速哈达马变换 (Fast Hadamard Transform) 广泛应用于 WCDMA 小区搜索的辅同步运算。本文给出了 FHT 法和传统直接相关法等效的证明, 论证了与直接相关法相比, FHT 大大降低了运算复杂度。并提出一种基于 FPGA 的辅同步实现方案, 实现中使用查找表法简化了相关值运算的复杂度, 节省了硬件资源。

关键词: WCDMA, 小区搜索, 快速哈达马变换, FPGA

1. 引言

小区搜索是移动通信中实现终端和基站之间同步的重要物理层过程。宽带码分多址接入 (WCDMA) 是全球使用最为广泛的 3G 标准之一, WCDMA 的初始小区搜索一般分为三个阶段: 主同步, 辅同步和第三步。首先, 在主同步阶段实现时隙同步, 然后辅同步在此基础上进行帧同步并确定扰码组号, 最后第三阶段确定主扰码号。经过小区搜索, 实现终端和基站的帧同步并得到基站的扰码号。

WCDMA 的一个无线帧包括 15 个时隙, 每个时隙有 2560 个码片。系统通过同步信道 SCH 实现下行同步。主同步信道 (Primary Synchronization Channel) 和辅同步信道 (Secondary Synchronization Channel) 在一个时隙的前 256 个码片传输, 见图 1。

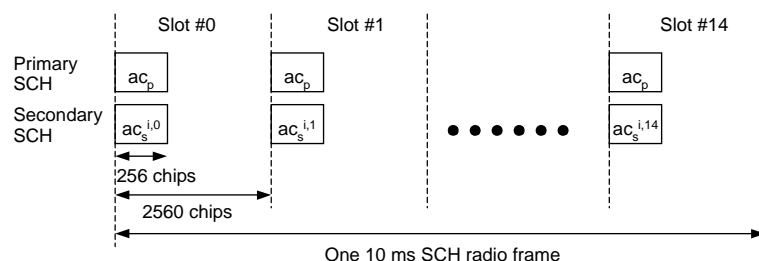


图1 WCDMA同步信道 (SCH) [1]

WCDMA 辅同步需要进行大量的相关运算, 一般使用传统的直接相关法实现。快速哈达马变换 (FHT) 通过蝶形运算简化相关计算的复杂度, 特别适用于 WCDMA 辅同步的简化实现^{[2][3]}。本文主要论述用 FHT 实现 WCDMA 辅同步的原理及其 FPGA 实现。

2. FHT 法实现 WCDMA 辅同步的原理

经过主同步得到时隙定时信息之后, 辅同步需要利用 SSCH 确定帧边界信息和扰码组号。使用 FHT 可以简化辅同步的运算复杂度, 下面首先给出 FHT 法在理论上和直接相关法等效的证明。

WCDMA 共有 512 个主扰码, 8 个一组分为 64 组。64 个扰码组号通过辅同步码子来标示。辅同步码字共有 16 种, 且长度为 256 的复同步码字, 实部和虚部相同。根据 3GPP 协议^[4], 辅同步码字为 Z 码和哈达马矩阵 (H_{256}) 特定 16 行的点积。Z 码定义如下:

$$b = \langle 1, 1, 1, 1, 1, 1, -1, -1, -1, -1, -1, -1, 1, 1, 1, 1 \rangle$$

$$z = \langle b, b, b, -b, b, b, -b, -b, b, -b, b, -b, -b, -b, -b, -b \rangle$$

哈达马矩阵通过如下的递推关系定义：

$$H_0 = (1)$$

$$H_k = \begin{pmatrix} H_{k-1} & H_{k-1} \\ H_{k-1} & -H_{k-1} \end{pmatrix}, \quad k \geq 1$$

$h_m(i)$ 和 $z(i)$ 分别表示 H_{256} 矩阵第 m 行和 z 码的第 i 个元素。则辅同步码子定义如下：

$$C_{ssc,k} = (1 + j) \times \langle h_m(0) \times z(0), h_m(1) \times z(1), h_m(2) \times z(2), \dots, h_m(255) \times z(255) \rangle$$

其中 $m = 16 \times (k - 1)$, $k = 1, 2, \dots, 16$

因为实部虚部完全相同，为了方便说明，以下论述中只考虑 $C_{ssc,k}$ 的实部。

经过主同步找到时隙头后，设 $X = \langle x_0, x_1, x_2, \dots, x_{255} \rangle$ 为一个时隙前 256 码片长度的信息序列。 $C_{ssc,1}, C_{ssc,2}, \dots, C_{ssc,16}$ 为 16 个 256 长度的辅同步码字。辅同步阶段需要将 X 与 16 个辅同步码子做相关，直接做相关计算量很大。可以采用 FHT (Fast Hadamard Transform) 来简化计算，具体的过程是用先用 Z 码对 X 做 16 长度的分段相关，再对分段后的 16 个累加值做 16 点的哈达马变换。证明如下：

仔细观察 H_{256} 和 H_{16} 发现

$$h_m(16 * n + j) = H_{256}(m, 16 * n + j) = H_{16}(k - 1, n)$$

$$m = (k - 1) * 16, \quad n, j = 0, 1, 2, \dots, 15, k = 1, 2, 3, \dots, 16$$

设 $corr(X, C_{ssc,k})$ 表示 X 和辅同步码子 $C_{ssc,k}$ 相关的结果，则：

$$corr(X, C_{ssc,k}) = \sum_{i=0}^{255} x(i) * C_{ssc,k}(i)$$

$$= \sum_{i=0}^{255} x(i) * z(i) * h_m(i)$$

$$= \sum_{n=0}^{15} \sum_{j=0}^{15} x(16 * n + j) * z(16 * n + j) * h_m(16 * n + j)$$

$$= \sum_{n=0}^{15} H_{16}(k - 1, n) \sum_{j=0}^{15} x(16 * n + j) * z(16 * n + j)$$

$$= \sum_{n=0}^{15} H_{16}(k - 1, n) * y(n)$$

其中 $y(n) = \sum_{j=0}^{15} x(16 * n + j) * z(16 * n + j)$, $n = 0, 1, 2, \dots, 15$ 为 X 序列与 z 码第 n 段的相关值。

定义 X 分别与 $C_{ssc,1}, C_{ssc,2}, \dots, C_{ssc,16}$ 相关的结果为 $corr(X, C_{ssc})$ ，由哈达马变换的定义可知：

$$corr(X, C_{ssc}) = \langle corr(X, C_{ssc,1}), corr(X, C_{ssc,2}), \dots, corr(X, C_{ssc,16}) \rangle^T$$

$$= H_{16} * Y = FHT_{16}(Y)$$

其中 $FHT_{16}(\cdot)$ 为 16 点的 FHT， $Y = \langle y(0), y(1), y(2), \dots, y(15) \rangle^T$ ，见图 2。

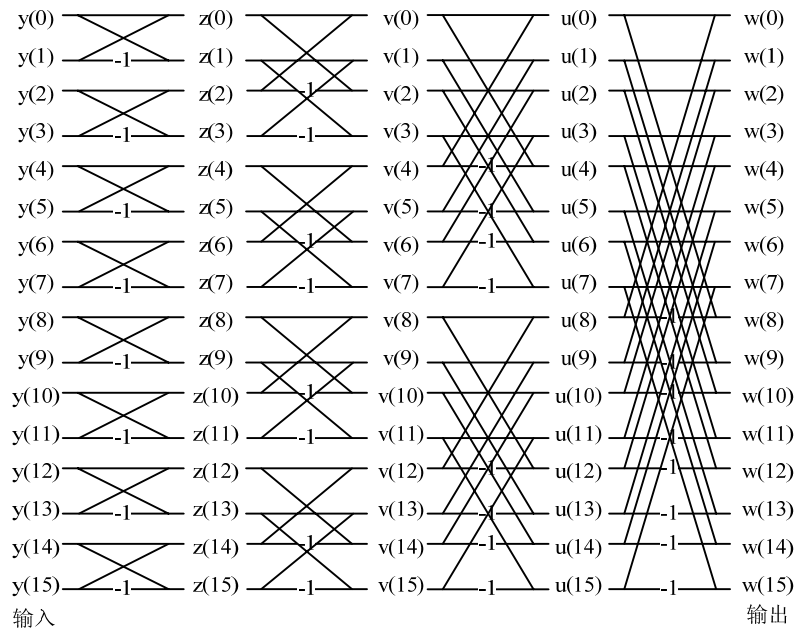


图 2 16 点 FHT 蝶形运算图

3. FHT 法和直接相关法的比较

下面从 X 分别与 $C_{ssc,1}, C_{ssc,2}, \dots, C_{ssc,16}$ 相关一次的计算量和存储量角度出发, 比较直接相关法和 FHT 法的优劣。

直接相关法需要储存所有 16 个 SSC 码字, 因此存储量为 $256 \times 16 = 4096 \text{bit}$, 因为要进行 16 次 256 长度的相关运算, 故需要 $256 \times 16 = 4096$ 次加法; FHT 法只需要存储 256bit 的 z 码和 256bit 的哈达马运算控制逻辑, 总计 512bit。先进行 16 段累加需要 $16 \times 16 = 256$ 次加法, 再进行 16 点的 FHT 需要 4 次蝶形运算, 见图 2 每次蝶形运算需要 8 次加法, 故 FHT 需要 $8 \times 4 = 32$ 次加法, 合计 $256 + 32 = 288$ 次加法。比较结果见表 1。

表 1 直接相关法和 FHT 法比较

	直接相关法	FHT 法	百分比 (FHT/直接相关)
存储量 (bit)	4096	512	12.5%
加法次数	4096	288	7.03%

可见 FHT 法在数据存储量和加法次数上都要优于直接相关法, 将其应用于 FPGA 的实现可以节省逻辑资源。

4. 基于 FPGA 的辅同步实现原理

每个时隙的前 256 码片都会和 $C_{ssc,1}, C_{ssc,2}, \dots, C_{ssc,16}$ 做相关, 经过 FHT 运算之后, 一帧数据得到 $16 \times 15 = 240$ 个相关值, 多帧累加后相关值个数不变。我们将这 240 个相关值储存在 FPGA 的 BRAM (Block RAM) 中, 为了便于理解, 给出相关值 ram_corr 的逻辑存储结果, 如图 3 所示。

存储相关值的 ram_corr 地址空间大小为 240, 按逻辑划分为 16 行 15 列, 每一列分别表示一帧的 15 个时隙分别和 16 个 SSC 码字相关的结果。

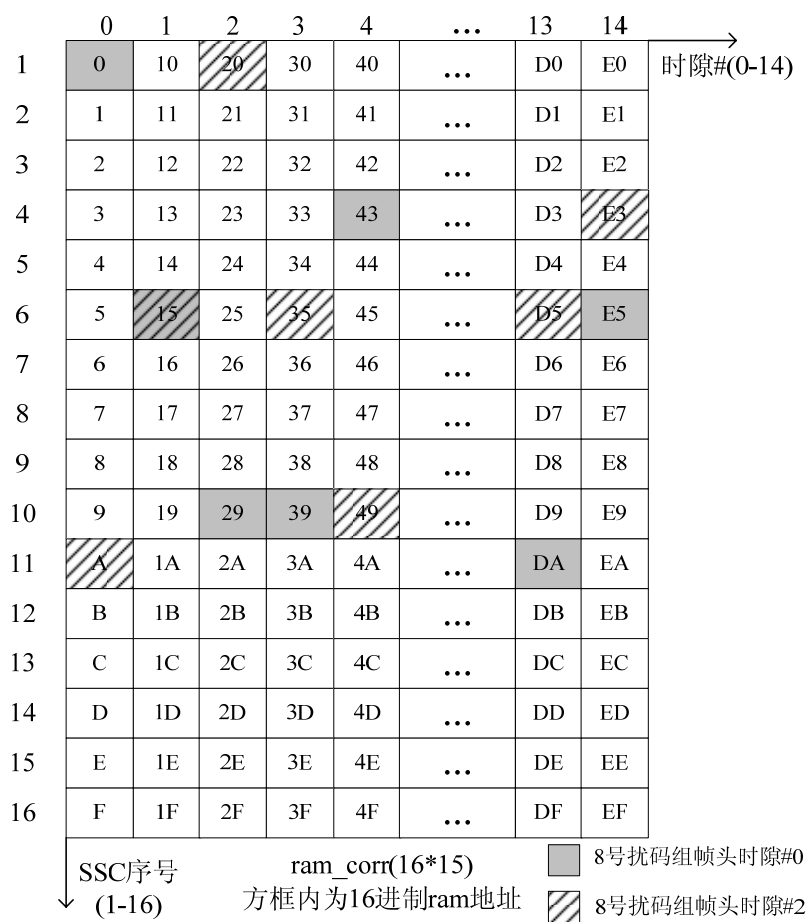


图3 16*15 相关值 ram

完成了所有时隙相关值的累加之后，需要根据 3GPP协议^[4]的扰码组SSC表（表 2）计算最终的相关值。辅同步模块运算流程如图 4 所示。

辅同步的功能在于找到扰码组号和帧边界信息。主同步之后得到 15 个时隙边界的位置，但由于不知道帧边界的位置信息，对于任何一个扰码组，我们需要遍历所有 15 个可能的帧头，因此总共需要 64*15=960 次相关值的累加和比较。下面以表中的 8 号扰码组为例说明如何通过查找扰码组表（ram_control）和相关值表（ram_corr）来计算最终的 960 相关值。8 号扰码组的 SSC 构成如下：

Group 8	1	6	10	10	4	11	7	13	16	11	13	6	4	1	16
---------	---	---	----	----	---	----	---	----	----	----	----	---	---	---	----

如用顺序 1,6,10,10,4,11,7,13,16,11,13,6,4,1,16 来做相关，则需要累加的 15 个相关值为图 3 中的灰色部分所示，对应帧头为时隙#0。通过地址产生逻辑，顺序 1,6,10,...,1,16 对应的相关值 ram 地址分别为 0,15,29,...,DA,E5。

如用顺序 10,10,4,11,7,13,16,11,13,6,4,1,16,1,6 来相关，则需要累加的 15 个相关值为图中的斜纹部分所示，对应帧头为时隙#2。通过地址产生逻辑，顺序 10,10,4,...,1,6 对应的相关值 ram 地址分别为 20,35,49,...A,15。

通过以上查找得到 15 个相关值后，再经过图 4 的 15 累加模块得到最终的相关值。

表 2 前 10 个扰码组的SSC表^[4]

Scrambling Code Group	slot number														
	#0	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	#14
Group 0	1	1	2	8	9	10	15	8	10	16	2	7	15	7	16
Group 1	1	1	5	16	7	3	14	16	3	10	5	12	14	12	10
Group 2	1	2	1	15	5	5	12	16	6	11	2	16	11	15	12
Group 3	1	2	3	1	8	6	5	2	5	8	4	4	6	3	7
Group 4	1	2	16	6	6	11	15	5	12	1	15	12	16	11	2
Group 5	1	3	4	7	4	1	5	5	3	6	2	8	7	6	8
Group 6	1	4	11	3	4	10	9	2	11	2	10	12	12	9	3
Group 7	1	5	6	6	14	9	10	2	13	9	2	5	14	1	13
Group 8	1	6	10	10	4	11	7	13	16	11	13	6	4	1	16
Group 9	1	6	13	2	14	2	6	5	5	13	10	9	1	14	10

通过 64*15 计数器控制 960 次相关值的累加和比较，最终得到最大相关值，并输出其对应的扰码组号和帧边界位置。由于引入了 16*15 的相关值表 (ram_corr)，我们无需再存储所有 960 个相关值，而是通过查表法将 15 个时隙的多帧相关值累加得到最终的 960 个相关值，并比较。这就可以通过一个加法器和比较器串行完成，从而节约了硬件资源，见图 4。

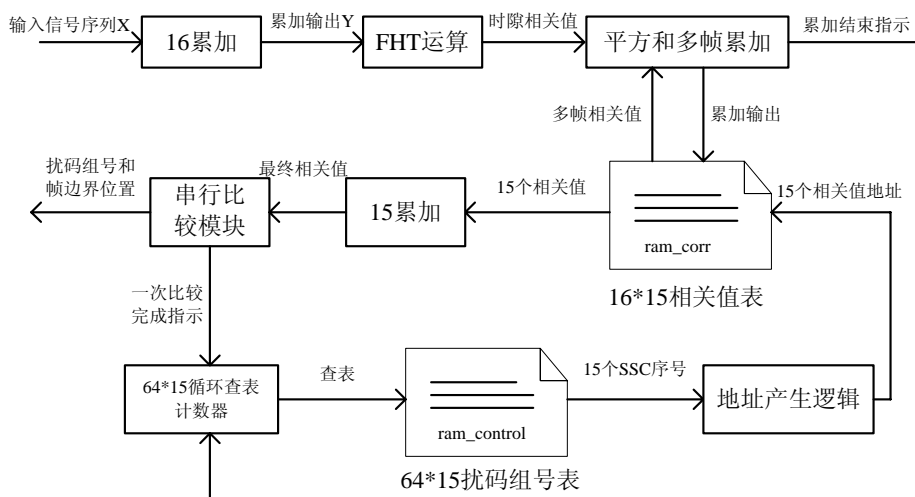


图 4 辅同步运算流程

Xilinx 一块的 BRAM 深度为 1024。为了节省资源将 64*15 的扰码组表、256bit 的 Z 码和哈达马运算逻辑放在同一块 BRAM ram_control 中。将 15*16 的相关值表和 I 和 Q 路 16 累加的 32 个中间结果放在 ram_corr 中，所有相关累加和 FHT 复用同一个加法器，同时由于平方相加操作，使用了一个乘法累加器 (DSP48)。在 xc4vsx35-ff668 器件上实现了辅同步模块，资源占用情况如表 3 所示。

表 3 辅同步占用资源情况

	辅同步占用资源	xc4vsx35 总资源	资源占用率
Slices	193	15,360	1.3%
BRAM	2	192	1.0%
DSP48	1	192	0.5%

5. 结束语

本文给出了 FHT 法进行 WCDMA 辅同步的证明，通过比较论证 FHT 法在存储量和计算复杂度上都要远优于传统的直接相关法。FHT 和查找表法的联合应用简化了 WCDMA 辅

同步模块的运算复杂度，在硬件实现时节省了大量的 FPGA 资源。最后给出了基于 FPGA 的辅同步实现方案和资源占用情况，具有一定的实用和参考价值。

参考文献

- [1] TS 25.211 v8.1.0 Technical Specification. Physical channels and mapping of transport channels (FDD) [S] . 3rd Generation Partnership Project (3GPP); Technical Specification Group (TSG) Radio Access Network (RAN); Working Group 1 (WG1). 2008-05
- [2] Sanat Kamal Bahl. Design and Prototyping a Fast Hadamard Transformer for WCDMA [R] . Proceedings of the 14th IEEE International Workshop on RSP'03, 2003
- [3] Alpaslan Demir and Donald Grieco. A robust algorithm for step 2 in 3G FDD WCDMA synchronization [J] . 2001,IEEE V TS 54th, (4) : 2462-2465
- [4] TS 25.213 v8.1.0 Technical Specification. Spreading and Modulation (FDD) [S] 3rd Generation Partnership Project (3GPP); Technical Specification Group (TSG) Radio Access Network (RAN); Working Group 1 (WG1). 2008-05.
- [5] Yi- P in EricWang and Tony Ottosson. Cell Search in WCDMA [J]. IEEE Select. Areas in Commun.,2000, 18 (8) : 1470- 1482
- [6] Xilinx. Virtex-4 User Guide UG070 (v2.3) [Z] Xilinx, 2007

FHT in WCDMA Secondary Synchronization and FPGA Based Implementation

Gu Tao

School of Information Engineering, Beijing University of Posts and Telecommunications,
Beijing PRC (100876)

Abstract

FHT(Fast Hadamard Transform) is widely adopted in WCDMA Secondary Synchronization. In this paper, we first demonstrate that FHT is equivalent to traditional correlation method with the same correlate result, however, with lower complexity. Then we propose a FPGA based implementation of WCDMA Secondary Synchronization. LUT (look-up-table) is used in correlate computation as a method to save FPGA resource.

Keywords: WCDMA, Cell Search, FHT, FPGA