

3.1 ModelSim 下载与安装

打开官方网站 <http://model.com>, 通过 Download → ModelSim SE → ModelSim SE Release Archive 路径选择合适的版本, 注册之后就可以下载了, 也可以使用教师提供的版本。

3.2 ModelSim 使用

1. 建立 Project 并添加文件

执行 File → New → Project 命令, 在出现的对话框中设置 Project Name 和 Project Location 选项, 在 Default Library Name 列表框中选择 work 选项, 单击 OK 按钮。

紧接着会弹出一个新窗口 Add items to the Project, 选择 Add Existing File 选项, 添加进所有要仿真的 VHDL 文件(可一次多选), 单击 OK 按钮。全部添加完毕后关闭 Add items to the Project 窗口。

之后在 Workspace 窗口的 Project 选项卡中也可以自行添加、删除 Project 中的文件。

需要注意, VHDL 文件存放的路径中不能包括中文, 否则目前版本会出错误。

在选定的工程文件夹中会自动生成以 Project Name 命名的 .mpf 文件, 之后打开这个 Project 只需要通过 File → Open → Project 命令选择这个 .mpf 文件即可。

通过菜单栏 Layout 可以在 NoDesign、Simulate、Coverage 三种窗口布局方式之间切换,当在屏幕上找不到某窗口时,可以从此处重置页面布局。

2. 编译文件

在 Workspace 窗口的 Project 选项卡中,在任一文件上右击,在弹出的快捷菜单中选择 Compile→Compile All 命令编译全部文件,之后在下方的 Transcript 窗口中会显示编译信息。

如果 Compile 结果中有 Error 或者 Warning,可在那条红色的提示信息语句上双击,在弹出的窗口上继续双击错误提示信息,便可以打开出现编译错误的代码位置。

编译全部通过后,就可以进行仿真了。每次修改过 VHDL 文件后,必须重新编译相应的文件才能进行仿真。

3. 进行仿真

在 Workspace 窗口切换到 Library 选项卡,展开 work 目录,双击要 simulate 的文件。或者在下方 Transcript 选项卡中输入“vsim 库名. 文件名”。

此时会在 Workspace 窗口中会出现 sim 选项卡(如果未找到,可通过 Layout 菜单切换到 NoDesign 布局)。

注意,在 sim 选项卡中双击 VHDL 文件所打开的窗口中代码是不能修改的,若要修改需要切换回 Project 选项卡中双击 VHDL 文件。

接下来通过执行 View→Objects 命令(本软件 6.0 之前的旧版本中为 Signals 命令)和 View→Wave 命令打开相应的两个窗口。或者在软件窗口下方的 Transcript 选项卡中输入“view 窗口名”。

在 Objects 窗口中右击,在弹出的快捷菜单中选择 Add to Wave→Signals in Region 命令,将所有信号添加到 Wave 窗口中。

在 Objects 窗口中选中某个 Signal,在上面右击,在弹出的快捷菜单中选择 Create Wave 命令,即可对其输入激励,如时钟、常量等。设置完毕后在 Wave 窗口中会多出相应的 Signal,可选中并右击,在弹出的快捷菜单中选择 Create/Modify Waveform 命令对输入激励进行修改。也可以通过在 Transcript 窗口中输入“force/unforce”命令来确定激励。例如

```
force clk 0 50, 1 100 - repeat 100
```

上面这条语句可赋给信号 clk 周期为 100ns 的 01 时钟信号,0 和 1 的比例为各占 50%。

通过在菜单 Simulation→Run 中选择相应条目,或在下方 Transcript 窗口中输入 Run、“Run 运行纳秒数”、“Run-all”、Restart 等命令,可以开始和重启模拟。

在程序暂停或是运行到断点时,要查看任意一个 Signal 或 Variable 的值,可以在 Wave 窗口中单击,从而出现黄色的垂直坐标线,拖动坐标线即可显示坐标处的时间与各数值。

在 Wave 窗口中在波形上双击,会打开 Dataflow 窗口,从中可以看到各 Signal 的连接情况,方便进行进一步调试。

如果不想以默认的二进制表示 Signal,可以在 wave 窗口内选定该 Signal,右击,在弹出的快捷菜单中选择 Signal Properties 命令,然后选择表示方法(如十进制表示 Decimal)。

4. 使用 Test Bench(测试平台文件)

自己编写 TestBench 文件相对复杂,但是比较灵活。

TestBench 文件就是一个简单的 VHDL 程序,它和需要仿真的实体对应,并且在程序中设置产生了相应的激励信号,也可以完成从文件读取数据、输出结果等功能。具体请查阅其他资料。

3.3 ModelSim 与 Quartus 联合使用

1. 实验前准备

首先确保 Quartus 与 ModelSim 均能单独正常使用。

2. 在 Quartus 中进行设置

执行 Tools→Options→General→EDA Tool Options 命令,在出现对话框右侧 ModelSim 窗格中设置 ModelSim 的存放路径。

使用模板编写 Test Bench: 执行 Processing→Start→Start Test Bench Template Writer 命令,然后根据下方 Messages 信息栏中显示的生成的 TestBench 文件位置打开该文件(与工程名称不同的 vht 类型文件),按照注释在相应位置补充填写需要测试的代码。

执行 Assignments→Settings→EDA Tool Settings→Simulation 命令,在出现对话框的右侧设置 Tool Name 为 ModelSim,在其下的 NativeLink settings 选项区域选中 Compile test bench 选项,选择刚刚创建好的 Test bench 文件,应注意填写 Test bench name 和 Top level module in the test bench 两项内容。

3. 进行仿真

执行 Tools→Run EDA Simulation Tool→EDA RTL Simulation 命令进行前仿。

执行 Tools→Run EDA Simulation Tool→EDA Gate Level Simulation 命令进行后仿。

前仿就是功能仿真,不考虑电路的门延迟与线延迟,重点在观察电路在理想环境下的行为与设计构想是否一致,所以仿真速度很快。前仿真结果正确,并不表示将来结果一定正确,但若前仿真结果不正确,则将来结果一定不正确。

后仿是在前仿的基础上进行的,需要借用 Quartus 生成几个文件(包括网标文件 .vo 和反标注文件 .sdo),考虑了电路的门延迟与线延迟,模拟结果更准确,但需耗费很多时间,所以建议在大型 Project 中,前仿正确后,再进行后仿。

具体 ModelSim 后仿的操作,请自行查阅相关资料学习。

4.1 海明码编码及校验

4.1.1 实验目的

- (1) 对容错技术有初步了解,理解掌握海明码的原理。
- (2) 掌握海明码的编码以及校验方法。

4.1.2 实验原理

数据位在传输过程中可能会发生误码,数据位由原来的 1 变成 0 或者 0 变成 1 的现象,这就需要对数据位进行检错、纠错。海明(Richard Hamming)码于 1950 年提出的,目前是被广泛采用的很有效的校验编码。它的特点是只要增加少数几个校验位,就能检测出多位出错,并能自动纠错。

海明码的实现原理是在数据中加入几个校验位,将数据代码的码距比较均匀地拉大,并把数据的每一个二进制位分配在几个奇偶校验(奇偶校验位是一个表示给定位数的二进制数中 1 的个数是奇数还是偶数的二进制数)组中。当某一位出错后,就会引起有关的几个校验位的值发生变化。这不但可以发现出错,还能指出是哪一位出错,为进一步自动纠错提供了依据。

假设校验位的个数为 r ,则它能表示 2^r 个信息,用其中的 1 个信息指出“没有错误”,其余的 $2^r - 1$ 个信息指出错误发生在哪一位。然而错误也可能发生在校验位,因此只有数据位 $k = 2^r - 1 - r$

个信息能用于纠正被传送数据的位数,也就是说要满足关系:

$$2^r \geq k + r + 1$$

若要能检测与自动校正一位错,并能发现两位错,此时校验位的位数 r 和数据位的位数 k 应满足下述关系:

$$2^{r-1} \geq k + r$$

按上述不等式,可计算出数据位 k 与校验位 r 的对应关系,如表 4.1 所示。

表 4.1 数据位 k 与校验位 r 的对应关系

k 值	最小的 r 值
1~3	4
4~10	5
11~25	6
26~56	7
57~119	8

若海明码的最高位号为 m ,最低位号为 1,即 $H_m H_{m-1} \cdots H_2 H_1$,则此海明码的编码规律通常是:

(1) 校验位与数据位之和为 m ,每个校验位 P_i 在海明码中被分在位号为 2^{i-1} 的位置上,其余各位为数据位,并按从低向高逐位依次排列的关系分配各数据位。

(2) 海明码的每一位码 H_i (包括数据位和校验位本身)由多个校验位校验,其关系是被校验的每一位位号等于校验它的各校验位的位号之和。

(3) 在增大合法码的码距时,所有码的码距应尽量均匀增大,以保证对所有码的检错能力平衡提高。

下面具体介绍对一个字节进行海明编码的实现过程。

只实现一位纠错两位检错,由前面的表可以看出,8 位数据位需要 5 位校验位,可表示为 $H_{13} H_{12} \cdots H_2 H_1$ 。

5 个校验位 $P_5 \sim P_1$ 对应的海明码位号分别为 H_{13} 、 H_8 、 H_4 、 H_2 和 H_1 。 P_5 只能放在 H_{13} 位上,因为 H_{13} 已经是海明码的最高位了,其他 4 位满足 P_i 的位号等于 2^{i-1} 的关系,其余位为数据位 D_i ,则有如下排列关系:

$$P_5 D_8 D_7 D_6 D_5 P_4 D_4 D_3 D_2 P_3 D_1 P_2 P_1$$

如上所述,每个海明码的位号要等于参与检验的几个校验位的位号之和的关系,可给出如表 4.2 所示的结果。

表 4.2 海明码位号与校验位位号的关系

海明码位号	数据位	校验位	参与校验的 校验位位号	海明码位号=校验位的海明码位号之和
1		P_1	1	$1=1$
2		P_2	2	$2=2$
3	D_1		1、2	$3=1+2$
4		P_3	4	$4=4$
5	D_2		1、4	$5=1+4$
6	D_3		2、4	$6=2+4$
7	D_4		1、2、4	$7=1+2+4$
8		P_4	8	$8=8$
9	D_5		1、8	$9=1+8$
10	D_6		2、8	$10=2+8$
11	D_7		1、2、8	$11=1+2+8$
12	D_8		4、8	$12=4+8$
13		P_5		$13=13$

由上述关系可以得出四个校验位各自与哪些数据位有关,进而得出关系表达式。

异或运算规则: $0 \oplus 1 = 1$ $1 \oplus 0 = 1$ $0 \oplus 0 = 0$ $1 \oplus 1 = 0$ 相同为 0,其余为 1。

$$P_1 = D_1 \oplus D_2 \oplus D_4 \oplus D_5 \oplus D_7$$

(注: 即 P_1 参与了 $D_1 D_2 D_4 D_5 D_7$ 数据位的校验,这些数据位异或后的结果即为 P_1)

$$P_2 = D_1 \oplus D_3 \oplus D_4 \oplus D_6 \oplus D_7$$

$$P_3 = D_2 \oplus D_3 \oplus D_4 \oplus D_8$$

$$P_4 = D_5 \oplus D_6 \oplus D_7 \oplus D_8$$

海明码距: 一个编码系统中任意两个合法编码间不同的二进制位数称为这两个海明码的码距,整个编码系统中任意两个码字的最小码距就是该编码系统的码距。(例如,1000 与 1001 之间前 3 位都为 100 而第四位则不同,1000 第四位为 0,1001 的第四位为 1,只有一位不同,则此海明码距为 1。又如 1010 与 0000 有 2 位数据位不同,则此海明码距为 2。)

各数据位形成 $P_i (i=1,2,3,4)$ 值时,不同数据位出现在 P_i 项中的次数是不一样的,使不同数据码的海明码的码距不等,并且有两位出错与一位出错分不清的问题。为此,还要补充一个 P_5 总校验位,使

$$P_5 = D_1 \oplus D_2 \oplus D_3 \oplus D_4 \oplus D_5 \oplus D_6 \oplus D_7 \oplus D_8 \oplus P_1 \oplus P_2 \oplus P_3 \oplus P_4$$

在这种安排中,每一位数据位,都至少出现在三个 P_i 值的形成关系中。当任一位数据码发生变化时,必将引起三个或四个 P_i 跟着变化,即合法海明码的码距都为 4。

如按如下关系对所得到的海明码实现偶校验,即:

$$S_1 = P_1 \oplus D_1 \oplus D_2 \oplus D_4 \oplus D_5 \oplus D_7$$

$$S_2 = P_2 \oplus D_1 \oplus D_3 \oplus D_4 \oplus D_6 \oplus D_7$$

$$S_3 = P_3 \oplus D_2 \oplus D_3 \oplus D_4 \oplus D_8$$

$$S_4 = P_4 \oplus D_5 \oplus D_6 \oplus D_7 \oplus D_8$$

$$S_5 = P_5 \oplus D_1 \oplus D_2 \oplus D_3 \oplus D_4 \oplus D_5 \oplus D_6 \oplus D_7 \oplus D_8 \oplus P_1 \oplus P_2 \oplus P_3 \oplus P_4$$

则得出的结果值 $S_5 \sim S_1$ 能反映 13 位海明码的出错情况。任何偶数位出错, S_5 一定为 0。

(1) 当 $S_5 \sim S_1$ 为 00000 时, 表明无错。

(2) 当 $S_5 \sim S_1$ 中仅有一位不为 0, 表明某一位校验位出错, 或三位海明码(包括数据位与校验位)同时出错。由于后一种出错的可能性要比前一种小得多, 认为就是一位出错, 出错位是该 S_i 对应的 P_i 位。

(3) 当 $S_4 \sim S_1$ 有不为 0 且 $S_5 = 0$ 时, 表明两位海明码同时出错, 此时只能发现这种错误, 而无法确定是哪两位错。

(4) 当 $S_4 \sim S_1$ 中仅有 1 位为 1 且 S_5 为 1, 表明是一位数据位出错; 当 $S_4 \sim S_1$ 中 2 位或 3 位为 1 且 S_5 为 1, 表明一位数据位出错, 出错位位号由 $S_4 \sim S_1$ 的编码值指明, 因此此时不仅能发现一位错(多个且是奇数位出错的概率很低, 这里不考虑), 而且能改正一位错, 即将出错的数据位变为其反码。

(5) 当 $S_4 \sim S_1$ 中 4 位为 1 且 S_5 为 1, 表明多位奇数个错, 超出海明码的纠正能力。

综上(2)和(4)所述, 一位错对应的 $S_5 \sim S_1$ 的值如表 4.3 所示。

表 4.3 海明码出错情况对照表

位号	13	12	11	10	9	8	7	6	5	4	3	2	1
数据位		D_8	D_7	D_6	D_5		D_4	D_3	D_2		D_1		
校验位	P_5					P_4				P_3		P_2	P_1
S_5	1	1	1	1	1	1	1	1	1	1	1	1	1
S_4	0	1	1	1	1	1	0	0	0	0	0	0	0
S_3	0	1	0	0	0	0	1	1	1	1	0	0	0
S_2	0	0	1	1	0	0	1	1	0	0	1	1	0
S_1	0	0	1	0	1	0	1	0	1	0	1	0	1

由此可得校验后的数据位表达式为:

$$D_1 = D_1 \oplus (S_1 \cdot S_2 \cdot \bar{S}_3 \cdot \bar{S}_4 \cdot S_5)$$

$$D_2 = D_2 \oplus (S_1 \cdot \bar{S}_2 \cdot S_3 \cdot \bar{S}_4 \cdot S_5)$$

$$D_3 = D_3 \oplus (\bar{S}_1 \cdot S_2 \cdot S_3 \cdot \bar{S}_4 \cdot S_5)$$

$$D_4 = D_4 \oplus (S_1 \cdot S_2 \cdot S_3 \cdot \bar{S}_4 \cdot S_5)$$

$$D_5 = D_5 \oplus (S_1 \cdot \bar{S}_2 \cdot \bar{S}_3 \cdot S_4 \cdot S_5)$$

$$D_6 = D_6 \oplus (\bar{S}_1 \cdot S_2 \cdot \bar{S}_3 \cdot S_4 \cdot S_5)$$

$$D_7 = D_7 \oplus (S_1 \cdot S_2 \cdot \bar{S}_3 \cdot S_4 \cdot S_5)$$

$$D_8 = D_8 \oplus (\bar{S}_1 \cdot \bar{S}_2 \cdot S_3 \cdot S_4 \cdot S_5)$$

例：海明码的生成。

已知数据位为

D_8	D_7	D_6	D_5	D_4	D_3	D_2	D_1
1	0	1	0	1	1	0	0

求加入校验位的海明码。

根据校验位公式：

$$P_1 = D_1 \oplus D_2 \oplus D_4 \oplus D_5 \oplus D_7$$

$$P_2 = D_1 \oplus D_3 \oplus D_4 \oplus D_6 \oplus D_7$$

$$P_3 = D_2 \oplus D_3 \oplus D_4 \oplus D_8$$

$$P_4 = D_5 \oplus D_6 \oplus D_7 \oplus D_8$$

$$P_5 = D_1 \oplus D_2 \oplus D_3 \oplus D_4 \oplus D_5 \oplus D_6 \oplus D_7 \oplus D_8 \oplus P_1 \oplus P_2 \oplus P_3 \oplus P_4$$

$$P_1 = 0 \oplus 0 \oplus 1 \oplus 0 \oplus 0 = 1$$

$$P_2 = 0 \oplus 1 \oplus 1 \oplus 1 \oplus 0 = 1$$

$$P_3 = 0 \oplus 1 \oplus 1 \oplus 1 = 1$$

$$P_4 = 0 \oplus 1 \oplus 0 \oplus 1 = 0$$

$$P_5 = 0 \oplus 0 \oplus 1 \oplus 1 \oplus 0 \oplus 1 \oplus 1 \oplus 1 \oplus 1 \oplus 1 \oplus 0 = 1$$

结合已知的数据位和计算所得的校验位 P ，得出如下结果：

13	12	11	10	9	8	7	6	5	4	3	2	1
P_5	D_8	D_7	D_6	D_5	P_4	D_4	D_3	D_2	P_3	D_1	P_2	P_1
1	1	0	1	0	0	1	1	0	1	0	1	1

即加入校验位的海明码为“1101001101011”。

由海明码的生成码写出海明码的校验位，并检查是否有错。

例：已知收到数据 1111001101011，请问该数据是否有错？是否能指出哪一位有错？是否能纠正该错误？

通过公式：

$$S_1 = P_1 \oplus D_1 \oplus D_2 \oplus D_4 \oplus D_5 \oplus D_7$$

$$S_2 = P_2 \oplus D_1 \oplus D_3 \oplus D_4 \oplus D_6 \oplus D_7$$

$$S_3 = P_3 \oplus D_2 \oplus D_3 \oplus D_4 \oplus D_8$$

$$S_4 = P_4 \oplus D_5 \oplus D_6 \oplus D_7 \oplus D_8$$

$$S_5 = P_5 \oplus D_1 \oplus D_2 \oplus D_3 \oplus D_4 \oplus D_5 \oplus D_6 \oplus D_7 \oplus D_8 \oplus P_1 \oplus P_2 \oplus P_3 \oplus P_4$$

可以得出 $S_5 \sim S_1$ 为 11011。

然后查海明码出错情况对照表 11011 对应的是 D_7 ，即海明码第 11 位，由于数据编码只有两种可能 1 或者 0。所以，把收到的数据第 11 位取反就会得到正确结果。

4.1.3 实验指导

首先根据信息位数,确定校验位数, $2^r \geq k+r+1$,其中 k 为信息为数, r 为校验位数,得出满足不等式的最小 r 值即为校验位数。

本实验采用海明(13,8)码,即编码前的数据 k 是 8bit,编码后为 13bit 即 $(k+r)$ 。

实验分为两部分:

- 输入待编码数据(8bit),查看编码结果;
- 输入校验数据,查看校验结果。

1. 信号分配

1) 编码部分(如图 4.1 所示)

信号 `byte_data[7:0]` 为原始数据。

信号 `byte_code[12:0]` 为编码结果。

2) 校验部分(如图 4.2 所示)

信号 `in_code[12:0]` 为待校验数据。

信号 `check_sum[4:0]` 为校验结果。

2. 接口与管脚分配

编码部分的管脚分配如图 4.1 所示。

1		<code>byte_code[12]</code>	Output	PIN_86	8	B8_NO
2		<code>byte_code[11]</code>	Output	PIN_87	8	B8_NO
3		<code>byte_code[10]</code>	Output	PIN_88	8	B8_NO
4		<code>byte_code[9]</code>	Output	PIN_105	7	B7_N1
5		<code>byte_code[8]</code>	Output	PIN_106	7	B7_N1
6		<code>byte_code[7]</code>	Output	PIN_109	7	B7_NO
7		<code>byte_code[6]</code>	Output	PIN_110	7	B7_NO
8		<code>byte_code[5]</code>	Output	PIN_111	7	B7_NO
9		<code>byte_code[4]</code>	Output	PIN_113	7	B7_NO
10		<code>byte_code[3]</code>	Output	PIN_114	7	B7_NO
11		<code>byte_code[2]</code>	Output	PIN_116	7	B7_NO
12		<code>byte_code[1]</code>	Output	PIN_117	7	B7_NO
13		<code>byte_code[0]</code>	Output	PIN_118	7	B7_NO
14		<code>byte_data[7]</code>	Input	PIN_168	5	B5_N1
15		<code>byte_data[6]</code>	Input	PIN_170	5	B5_NO
16		<code>byte_data[5]</code>	Input	PIN_171	5	B5_NO
17		<code>byte_data[4]</code>	Input	PIN_173	5	B5_NO
18		<code>byte_data[3]</code>	Input	PIN_174	5	B5_NO
19		<code>byte_data[2]</code>	Input	PIN_175	5	B5_NO
20		<code>byte_data[1]</code>	Input	PIN_177	5	B5_NO
21		<code>byte_data[0]</code>	Input	PIN_178	5	B5_NO

图 4.1