《数字集成电路设计与实践》芯片测试报告

1.1 功能测试

1.1.1 结果截图

具体结果可见附件视频，下节选部分典型截图：

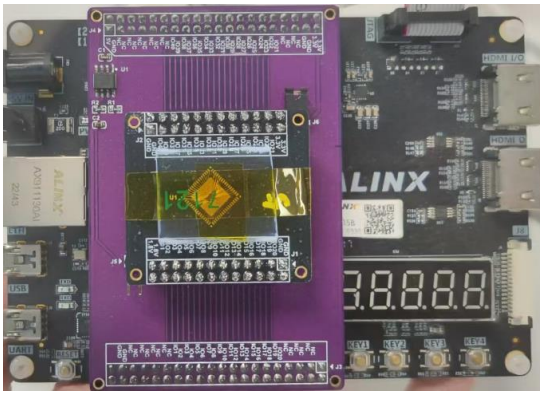

图 1 芯片与转接板、FPGA 的连接（进行功能测试时 J5、J6 跳线帽均连接）

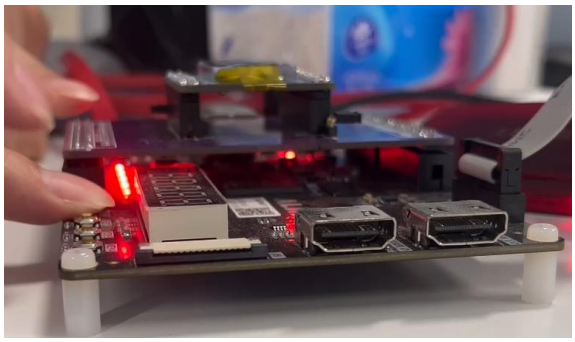

图 2 基础指令测试（最左侧一个 LED 亮说明正确执行）

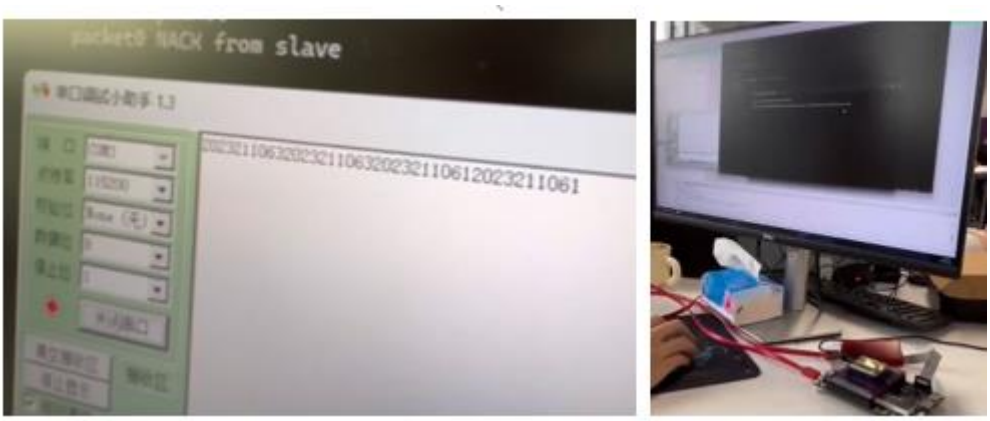

图 3 扩展 sID 指令测试（左图为串口发回的结果，右图为测试现场）

1.1.2 正常工作最高频率

为了实现频率变化，在 FPGA 转接模块中添加了一个 MMCM IP 用于时钟分频，经过多轮测试得到以下结果：

芯片支持的最高频率为 217MHz，再高 uart 时序就不满足了，程序烧不进去。基本指令测试选择的 case 为 inst_simple，以及额外测试了 sID，能正常执行，但是该情况下 inst_add/inst_div 等路径较长的 case 不能正确执行。

1.2 性能测试

1.2.1 测试环境与芯片连接

测试时室温 20℃，湿度 60%。

连接如下图，J5 跳线帽取下后使用直流电源供电，IO 口的 3.3V 依然采用板载供电。一个万用表作为电流表测量动态电流，一个作为电压表测量加到芯片两端的电压。

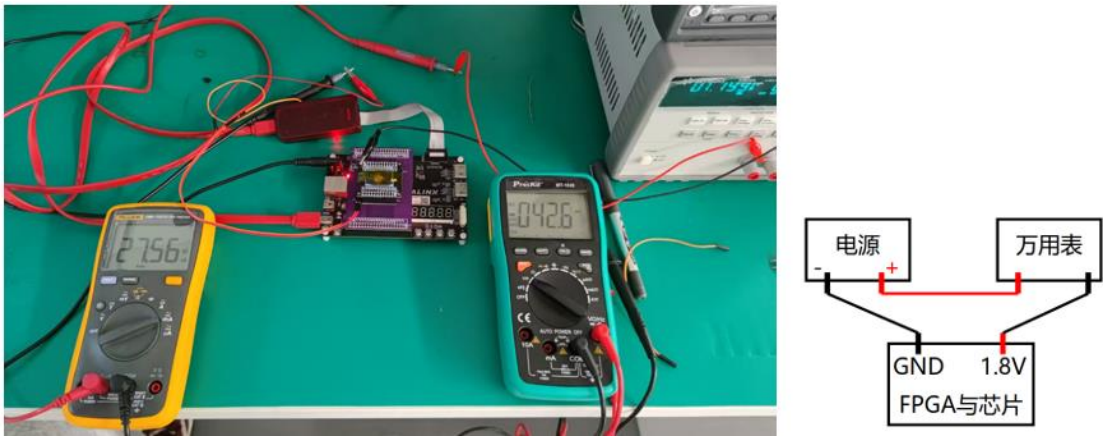

图 4 仪器连接实物与示意图（左图为实物，右图为示意）

1.2.2 性能、功耗曲线

测试方案为在给定电压下，测量芯片能正常工作（正常工作定义为能烧进程序且能够正确运行 inst_simple case）的最高频率，得到最高频率后读出当前的电流与加给核的电压，最终得到如图 5 曲线。

测量时电压从 1.8V 往下减少，在每个电压值下测得芯片正常工作的最高频率与工作时的动态电流与芯片供电口两端的电压（也就是实际的电压），之后再计算功耗、绘制曲线。实际上，电压越高、频率越大，芯片分得的电压比例越小，猜测是导线连接部分的电阻电感分压导致的。

可以看到，在 1.8V 时内核电压为 1.43V，能够运行在 166MHz；当电压降到 1.3V(1.153V)后在 50MHz 的频率下也不能再烧写程序进去。计算得到最佳效率点是 1.6V-140MHz。

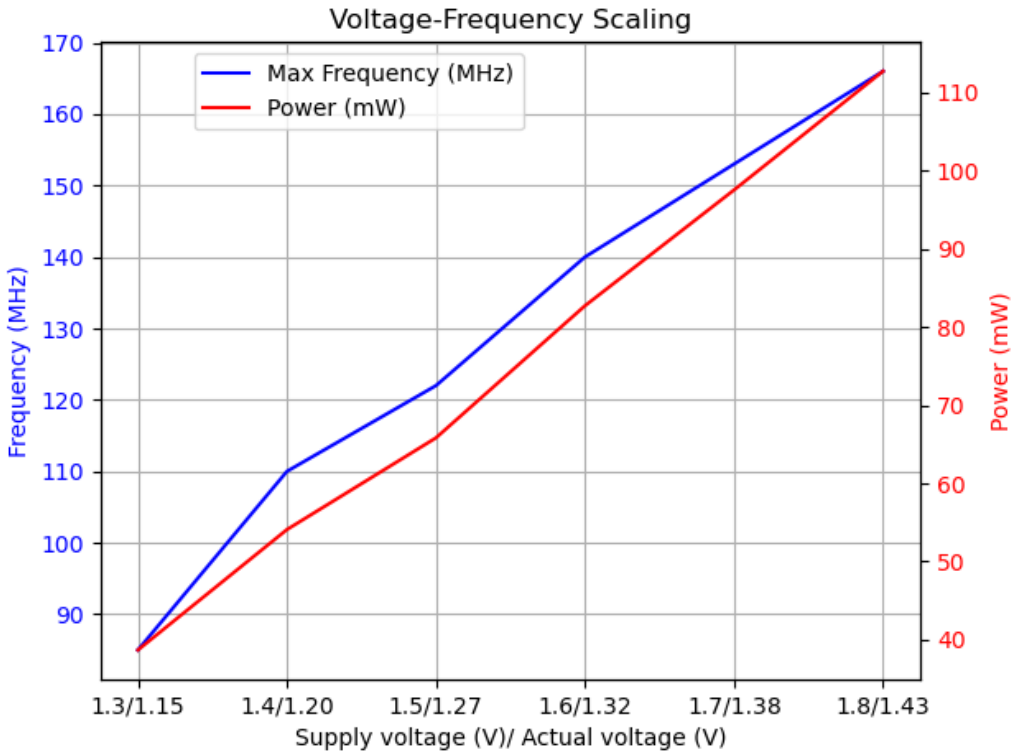

图 5 性能、功耗曲线图

1.3 内核部分改进

1.3.1 静态分支预测

添加了静态分支预测，采用的方法是 BTFN 预测，即对于向后的跳转预测为跳，向前的跳转则预测为不跳. 向后的跳转是指跳转的目标地. 依据是在实际的汇编程序中向后分支跳转的情形要多于向前跳转的情形，譬如常见的 for 循环生成的汇编指令往往使用向后跳转的分支指令；对于 jal 指令，提前译码消除了流水线气泡

提前译码模块见 pre_id.v，分支预测策略见 bpu.v

对于 simple 测试用例如下图所示，效率提高了 10.66%；且分支预测逻辑较为简单，在取指模块中也不会影响关键路经，目测不会对整体时钟频率造成影响。

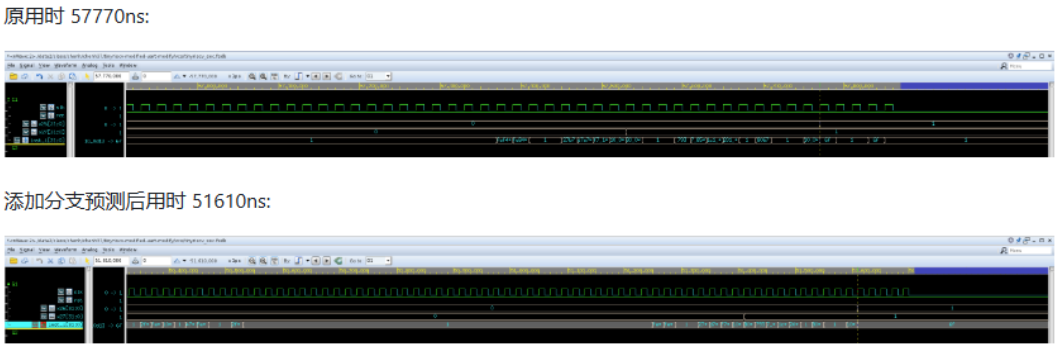

图 6 分支预测优化效果

1.3.2 优化流水线

修改内核流水线为 4 级，增加了 write back 阶段，并设计了部分前递逻辑防止数据冒险。有助于提高指令吞吐量与时钟频率

1.3.3 访问外设的握手优化

Read Temperature 指令指令功能可以等效为一次访存，但由于 i2c 读取温度读数有延迟，需要为 i2c 的访问增加握手信号：在 ex.v 中运行该指令会向外设发送 req 信号，等待过程中暂停流水线；ack 到达握手成功后重启流水线，这样就可以防止多次执行该指令造成的数据丢失

1.3.4 Uart 自适应波特率

为了能够在任意时钟频率下通过 uart_debug 模块实现程序更新，为 uart 模块设计了自适应波特率模块：在初始化时上位机向 uart 模块发送一次 0x55 即可完成波特率 115200 下的 uart 分频系数设定，之后即可通过上位机更新 rom（当然也可以手动计算波特率）

在此设计下，禁用了 uart_debug 模块的波特率更新，防止冲突

在不同的时钟频率下软件需要使用 i2c 和 uart 的话只能通过软件(写分频寄存器)来进行配置