

零成本快速完成 SOC 概念验证

JEFF MILLER, MENTOR GRAPHICS 产品营销和策略部



A M S D E S I G N & V E R I F I C A T I O N

W H I T E P A P E R

www.mentor.com

场景

您的公司提供基于模拟 / 混合信号 (AMS) 和传感器的 IC，而您的优质客户希望您创建一个包含数字处理器的片上系统 (SoC)，如图 1 所示。您运用数字处理器的经验不足，但您需要在接下来的几天内快速提供一个概念验证，向客户证明这款新 IC 的可行性。而且，您的预算非常之少。

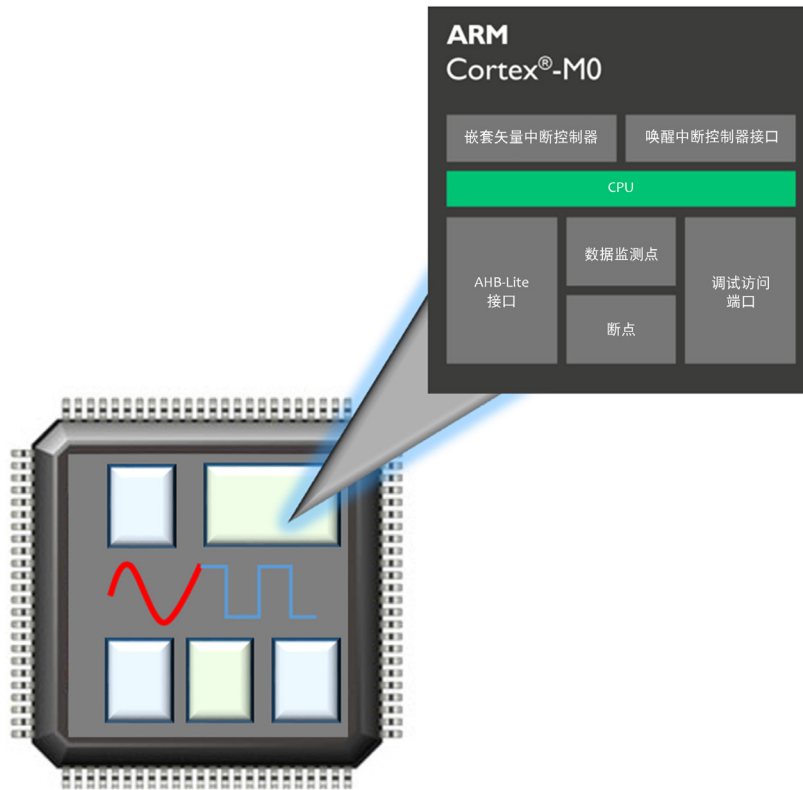


图 1: 场景 —— 扩展 AMS 设计以包含处理器。

由于缺少资金，您需要尽可能降低非重复性工程 (NRE) 成本。在本文中，NRE 的工作定义为 IP 和 EDA 工具作业的成本。工资是重复性的，因此概念验证所花的时间不包括在 NRE 中，但不管怎样，您只有几天时间来完成概念验证。如何能快速向客户提供概念验证，同时又不产生 NRE 成本呢？

ARM® DESIGNSTART™

ARM 认识到传感器和混合信号公司、初创公司以及小型团队对创建定制 SoC 有特殊要求，因此提供了 [DesignStart](#) 门户（图 2）以便设计人员能够快速、轻松、免费试用各种精选 ARM 产品。此外，MentorGraphics 提供了 Tanner EDA 设计工具的免费评估版，而 ARM 提供了其认可的设计合作伙伴来协助进行 SoC 开发。

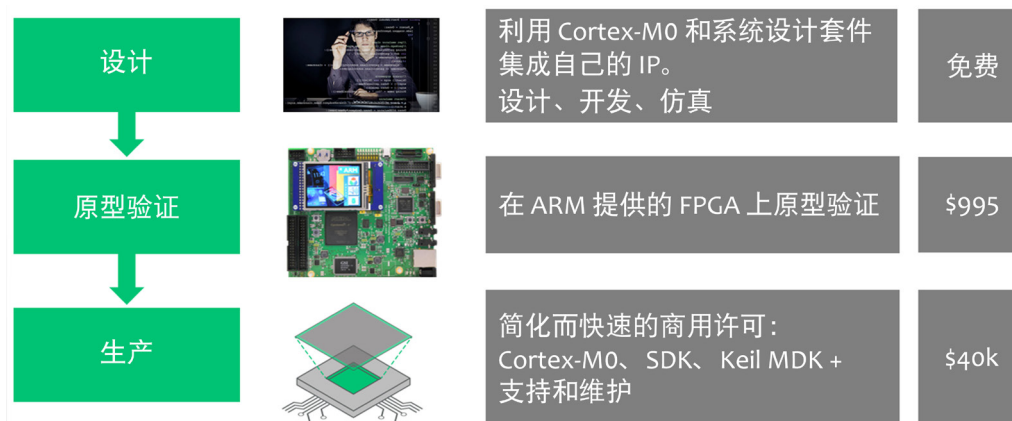


图 2：ARM DesignStart 门户（来源：ARM）。

针对您的项目，该门户提供有 ARM Cortex®-M0 处理器，您可以免费下载并将其用于设计和仿真。这是您的快速概念验证项目的理想解决方案。ARM Cortex-M0 是一款小尺寸、低功耗的 32 位处理器（图 3）。



图 3：ARM Cortex-M0 处理器（来源：ARM）。

此处理器广泛用于工业中对成本敏感的设备，主要特性如下：

- 内置低功耗特性 —— 例如休眠、深度休眠和状态保留低功耗模式
- 确定性指令执行时序 —— 指令和中断具有固定时序，中断处理自动执行
- 罕见的代码密度 —— 精简代码，代码少于 8/16 位器件
- 占用面积非常小 —— 仅有 12k 个门，用 8 位处理器的门数量实现 32 位处理
- 轻松快捷的开发 —— 仅有 56 条指令和一个 AHB 总线接口，可以快速掌握整个 Cortex-M0 指令集及其对 C 语言友好的架构

如果您刚刚才开始使用 Cortex-M0 处理器，您可访问 [ARM 开发人员](#)，了解更多信息。

TANNER EDA 解决方案

Mentor graphics 提供有 Tanner EDA 工具的 30 天[免费评估版](#)，您可以利用它来设计概念验证 SoC 并进行仿真。Tanner EDA 是一个高度集成的、完整的前端到后端的数模混合信号 IC 设计解决方案。针对您的概念验证项目，您可以利用 S-Edit 创建 AMS 电路图，集成 ARM Cortex-M0 内核，并利用 T-Spice 和 ModelSim 仿真整个设计（图 4）。验证概念之后，您可以使用全套设计工具进行版图布局并验证设计（不在本文讨论范围之内）。



图 4: Tanner 设计和仿真流程。

项目准备工作包括如下步骤：

1. 注册 [DesignStart](#) 以便免费使用 Cortex-M0 处理器。
2. 获得许可之后，将 Cortex-M0 设计套件下载到项目区域。
3. 注册以获得 Tanner EDA 工具的 30 天免费评估版。
4. 下载 Tanner EDA 工具并设置许可证。

现在，一切准备就绪，您可以开始概念验证设计了。

示例设计

了解概念验证主要步骤的最佳办法是完成一个示例设计（图 5）。

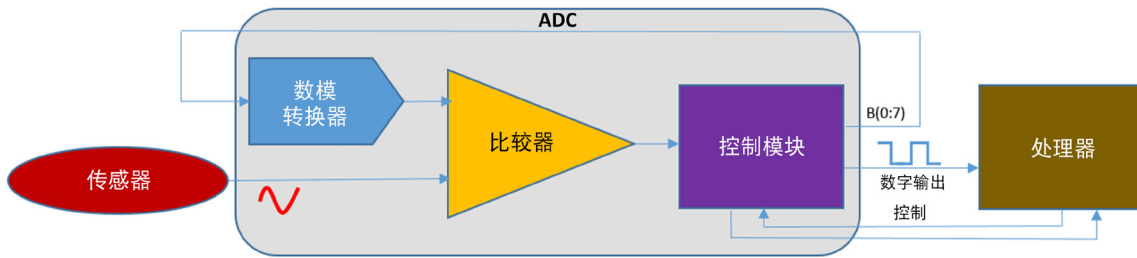


图 5: 示例设计——一个传感器驱动一个与处理器集成的 ADC。

我们的示例设计显示，一个模拟传感器连接到了一个 8 位模数转换器 (ADC)，而我们希望将 ADC 连接到 ARM Cortex-M0 处理器。由于这是一个概念验证，您无需大费周章地编写处理器上运行的软件程序。相反，您只需通过仿真确认来自控制模块的数字串行输出能与处理器正确通信即可。因此，该项目应把重点放在 AMS 设计与处理器的正确接口上。

设计和仿真流程

ARM Cortex-M0 [DesignStart 设计套件](#)（图 6）提供了带周边元件的预集成处理器子系统。利用示例设计，您可以将控制模块连接到 AHB2APB 接口，后者可连接到 AHB 接口。ARM AMBA® 先进高性能总线 (AHB) 规范是处理器的主要系统总线，而 AMBA 先进外设总线 (APB) 规范则可用于连接周边元件。该系统经您（而非仿真器）模糊处理过，由具 ARM Cortex-M0 功能的 Verilog 源文件组成。

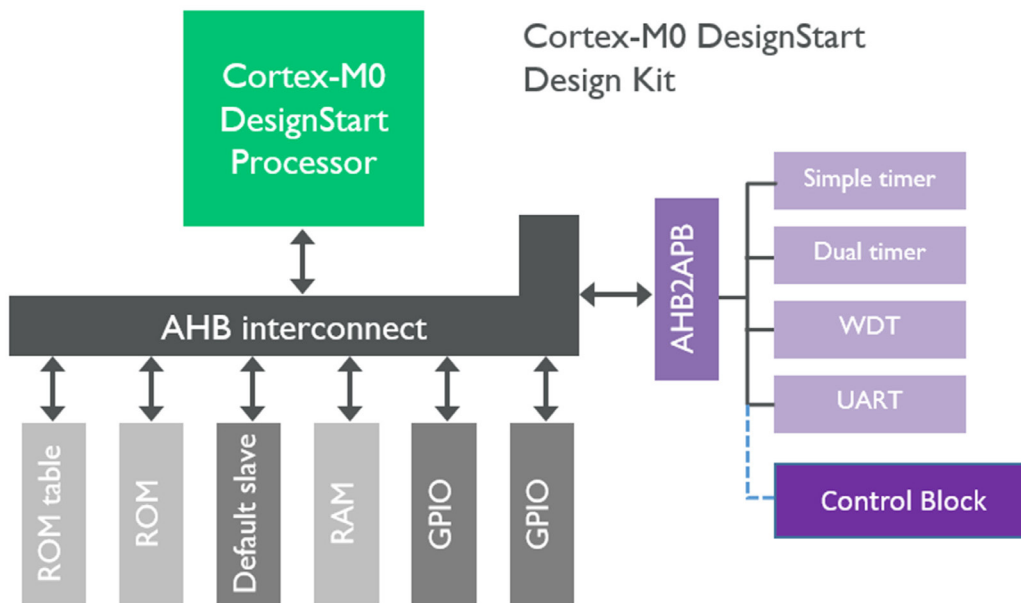


图 6: 控制模块连接到 ARM Cortex-M0 处理器（来源：ARM）。

创建接口

控制模块是一个描述元器件行为的 Verilog 模块（图 7）。您可以在 S-Edit 中利用文本编辑器创建此描述。

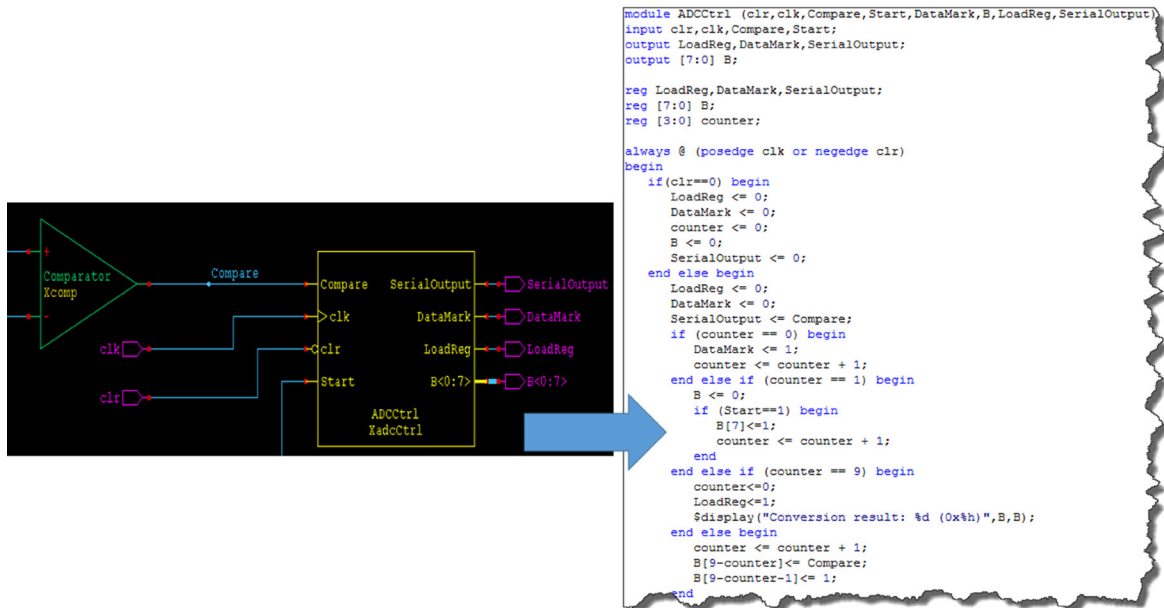


图 7：控制模块的 Verilog 描述。

为将控制模块 (ADCCtrl) 连接到子系统总线，您需要创建一个可将控制模块的输入和输出映射到 APB 的 Verilog 模块。这意味着您需要了解 APB 标准文档（可以从 ARM 免费[下载](#)获得）。在一个文本文件内定义一个模块（图 8），内容如下：

1. 定义 APB 输入和输出。
2. 指定您需要从控制模块连接的设计输入和输出。
3. 指定行为代码（图 9）中需要使用的所有设计信号。
4. 建立控制模块、设计 I/O 与此接口模块之间的端口映射。

```
module apb_adc8 (  
  input wire      PCLK,      // PCLK for timer operation  
  input wire      PCLKG,    // Gated clock  
  input wire      PRESETn,   // Reset  
  
  input wire      PSEL,     // Device select  
  input wire [11:0] PADDR,   // Address  
  input wire      PENABLE,  // Transfer control  
  input wire      PWRITE,   // Write control  
  input wire [31:0] PWDATA,  // Write data  
  
  output reg [31:0] PRDATA,  // Read data  
  output reg      PREADY,   // Device ready  
  output reg      PSLVERR,  // Device error response  
  
  input wire      ADC_COMPARE,  
  output wire [7:0] ADC_DAC_DATA  
  
);  
  
reg Start;  
wire DataMark, LoadReg, SerialOutput;  
  
reg idle;  
  
ADCCtrl u_adcctrl (.clr(PRESETn),.clk(PCLK),.Compare  
(ADC_COMPARE),.Start(Start),.DataMark(DataMark),.B  
(ADC_DAC_DATA),.LoadReg(LoadReg),.SerialOutput(SerialOutput));  
  
always @(posedge PCLK or negedge PRESETn)  
begin  
  if(!PRESETn)  
    begin
```

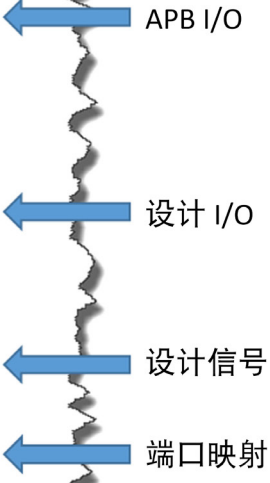


图 8: 定义接口和端口映射。

接着，您可以在同一文件（图 9）中定义 I/O 和信号的实际行为，包括就绪状态、何时等待、以及如何发送和接收数据。

```

always @(posedge PCLK or negedge PRESETn)
begin
  if(!PRESETn)
  begin
    PRDATA <= {32{1'b0}};
    PSLVERR <= 1'b0;
    Start <= 1'b0;
    idle <= 1'b0;
    PREADY <= 1'b0;
  end
  else
  begin
    Start <= 1'b0;
    PRDATA <= {32{1'b0}};
    if(PENABLE && PSEL && ~DataMark && ~PWRITE && ~idle)
    begin
      Start <= 1'b1;
      idle <= 1'b1;
    end
    if(PENABLE && PSEL && DataMark && ~PWRITE)
    begin
      PRDATA[7:0] <= ADC_DAC_DATA;
      PREADY <= 1'b1;
    end
    if(~PSEL || ~PENABLE)
    begin
      idle <= 1'b0;
      PREADY <= 1'b0;
    end
  end
end
endmodule

```

图 9: 总线接口行为。

插入 MCU 系统

接下来，我们需要将外设连接到 Cortex-M0 系统。MCU 系统模块为此提供了数个 APB 扩展端口。对于本例，我们选择端口 15。在代表该系统的 Verilog 代码的定义部分中，增加 ADC 所用的信号（图 10）。

```

wire [31:0] ADC8_PWDATA, ADC8_PRDATA, ADC8_PADDR;
wire ADC8_PSEL, ADC8_PENABLE, ADC8_PWRITE, ADC8_PREADY, ADC8_PSLVERR;

```

图 10: 定义 ADC 信号。

在同一文件的 APB 子系统配置部分中，将 ADC 信号连接到 APB 上对应 APB 端口 15 的信号。最后对 APB 接口模块进行实例化（图 11）。


```

apb_adc8 u_adc8 (
  .PCLK(PCLK),
  .PCLKG(PCLKG),
  .PRESETn(PRESETn),
  .PSEL(ADC8_PSEL),
  .PADDR(exp_paddr),
  .PENABLE(exp_penable),
  .PWRITE(exp_pwrite),
  .PWDATA(exp_pwdata[31:0]),
  .PRDATA(ADC8_PRDATA),
  .PREADY(ADC8_PREADY),
  .PSLVERR(ADC8_PSLVERR),
  .ADC_COMPARE(adc_compare),
  .ADC_DAC_DATA(adc_dac_data)
);

```

图 11: APB 接口模块中的 ADC 实例化。

编写软件

通过接口定义将控制模块连接到 Cortex-M0 子系统之后，您需要编写一个在处理器上运行的测试程序。该程序必须编译到 Cortex-M0 上运行，因此您可以使用 ARM Keil® MDK-Lite，它是一个用于创建、编译和调试程序的软件开发解决方案。有关如何获得评估版的详细信息，请参阅[下载](#)页面。或者，您可以使用 [DesignStart 包](#) 中附带的 Keil MDK 90 天许可专业版。

快速创建软件测试程序的最简单办法是复制和修改 DesignStart 设计套件中包含的 *hello.c* 文件（图 12）。

```

#include <stdio.h>
#include "uart_stdout.h"
#define ADCDATA (* (volatile unsigned char*)0x4000F000L);

int main (void)
{
  // UART init
  UartStdOutInit();
  printf("Hello world\n");

  int v = ADCDATA;
  printf("Reg value is %d\n", v);

  if(v==215)
    printf("*** TEST PASSED **\n");
  else
    printf("*** TEST FAILED **\n");
  // End simulation
  UartEndSimulation();

  return 0;
}

```

图 12: 测试软件程序。

软件程序在仿真 Cortex-M0 内部操作 ADC。ADCDATA 已被定义及设置为基准，即 APB 端口 15 的存储器映射地址的十六进制值。程序通过 UART 模块在仿真器中执行 *printf* 语句。代表传感器信号的 ADC 输入电压设置为 1.8V。ADC 参考电压设置为 2.2V。其与 215 次计数的 ADC 输出值 $((1.85V/2.2V) * 256)$ 相对应。程序会检查是否仿真了 215 次计数。若是，则测试成功。若否，则测试失败。

利用 Keil MDK-Lite 工具将此程序编译到 ModelSim *work* 目录中。

设计仿真

S-Edit 创建 Verilog-AMS 网表，然后传递给 T-Spice。如图 13 所示，T-Spice 会自动分割网表，从而将设计分为模拟仿真和 ModelSim 中的数字仿真。

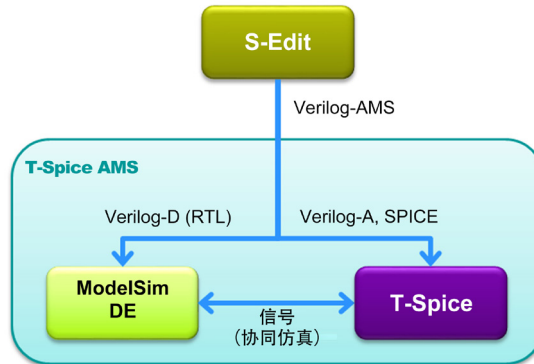


图 13: 模拟数字仿真划分。

两种仿真器均被自动调用，而在仿真期间，信号值在仿真器之间来回传送。这意味着不论您使用哪种设计实施语言（SPICE 或 Verilog），您只需从 S-Edit 运行仿真，设计就会被自动拆分到各个仿真器中。之后，您可以使用 ModelSim 或 T-Spice 波形查看器查看结果。

示例设计包含一个测试平台，其可提供电压源和时钟，并可捕捉 I/O 值以显示波形（图 14）。对于此概念验证，用固定不变的 1.8V 模拟电压代替传感器信号。

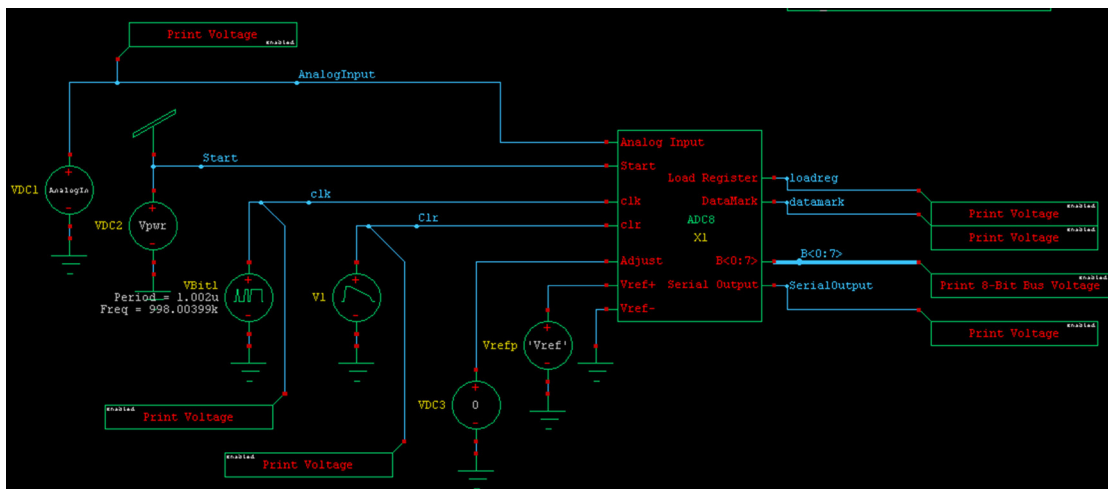


图 14: 设计测试示例。

8 位 ADC 通过逐次逼近法将传感器的模拟信号转换为离散数字表示以输入到处理器中。控制模块给出一个较高或较低的“猜测”，而比较器将模拟输入与数模转换器 (DAC) 的输出相比较，如图 15 中的波形所示。

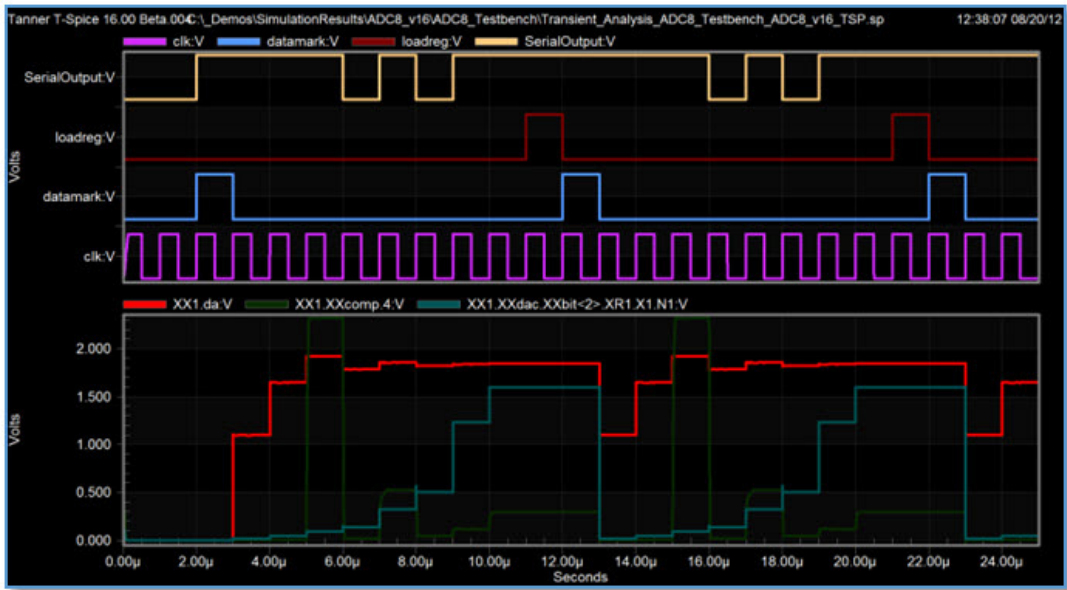


图 15: 显示逐次逼近的波形。

如果仿真成功，您将看到软件程序运行所得出的控制模块的真实仿真状态。

总结

该概念验证项目的重点是让 AMS 设计精密连接到 [Cortex-M0 处理器](#)，以向最终客户证明设计可行。您可以在不花费任何 NRE 成本的情况下快速完成此项目：

项目	成本
ARM Cortex-M0 DesignStart 设计套件	\$0
Tanner EDA 工具集 (30 天评估版)	\$0
AMBA APB 规范	\$0
Keil MDK-Lite 软件开发解决方案	\$0
非 NRE: 创建和验证概念设计	天数 X 每天费用

后续步骤

有了概念验证，您只需创建一个演示，甚至做几张幻灯片就可以向客户介绍您的项目。客户满意之后，下一步便是实施 SoC。这就要求您购买流程中用到的工具，并且购买含有 ARM 简化快速许可证的 Cortex-M0 及设计套件。然后，您需要完成后端实施流程（图 16），并将设计交给您选择的晶圆代工厂。



图 16: Tanner EDA 后端实施流程。

如需最新信息，请致电联系我们，或者访问：

www.mentor.com

©2017 Mentor Graphics Corporation，保留所有权利。本文档包含 Mentor Graphics Corporation 的专有信息，只能由原始接收者出于内部商业目的的全部或部分复制本文档，前提是在所有副本中都包含此完整声明。接受本文档即表示接收者同意采取一切合理措施，防止未经授权使用这些信息。本文档中提及的所有商标属于其各自所有者。

公司总部
 Mentor Graphics Corporation
 8005 S.W. Boeckman Road
 Wilsonville, Oregon 97070 USA
 电话: +1-503-685-7000
 传真: +1-503-685-1204
销售和产品信息
 电话: +86-21-6101-6301
 sales_info@mentor.com

上海
 明导 (上海) 电子科技有限公司
 上海市浦东新区杨高南路 759 号
 陆家嘴世纪金融广场 2 号楼 5 楼
 邮编: 200127
 电话: +86-21-6101-6301
 传真: +86-21-5047-1379

北京
 明导 (上海) 电子科技有限公司
北京办事处
 北京市南礼士路 66 号
 建威大厦 1512 室
 邮编: 100045
 电话: +86-10-5930-4001
 传真: +86-10-6808-0319

深圳
 明导 (上海) 电子科技有限公司
深圳办事处
 深圳市福田区金田路 3088 号
 中洲大厦 24 楼 2401 室
 邮编: 518040
 电话: +86-755-8282-2700
 传真: +86-755-8826-7750

