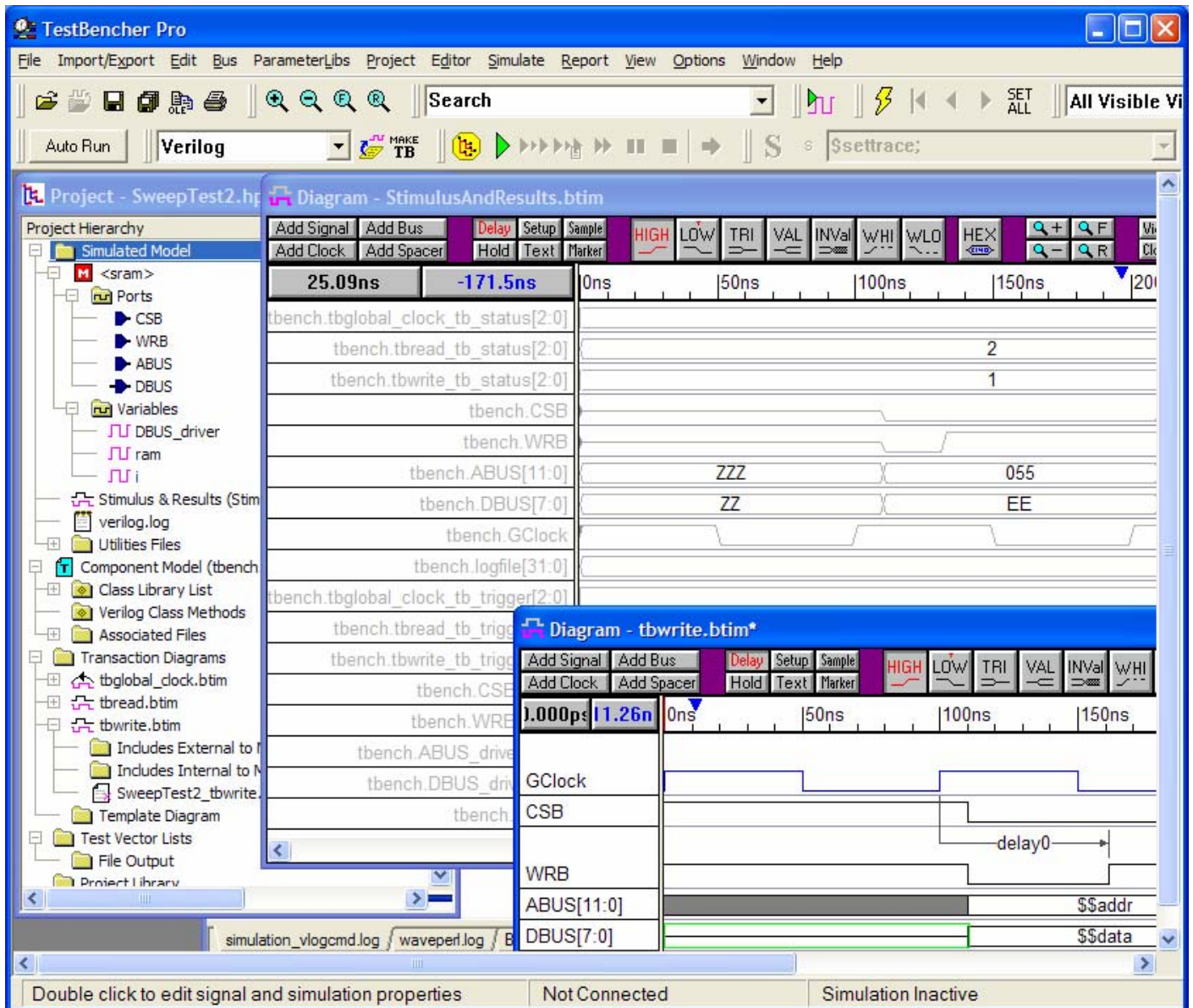


TestBencher Pro... sehr praktisch...



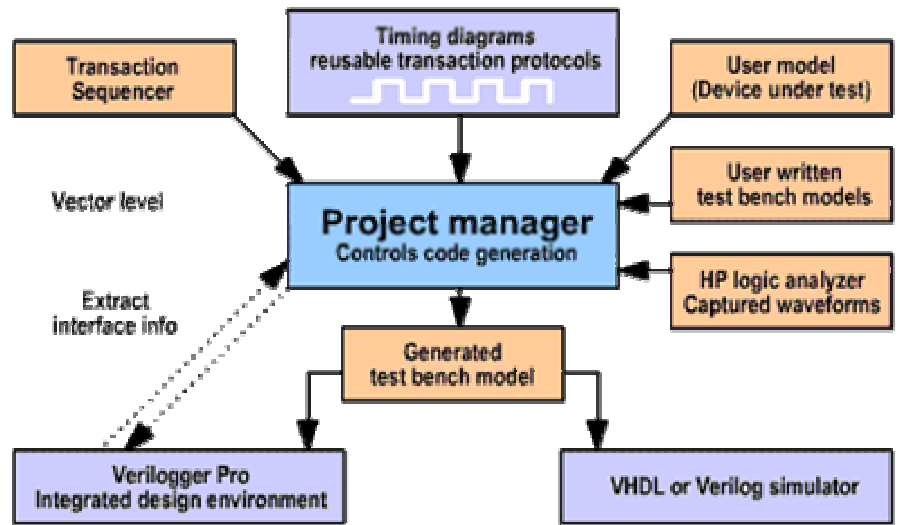
...und sehr gut für Ihren HDL Simulations Flow

TestBencher Pro

bietet Ihnen eine grafische Oberfläche zur Entwicklung von Testbenches in der HDL Ihrer Wahl und zur Steuerung kompletter Simulationen einschließlich der Auswertung von Simulations-Ergebnissen sowie die Steuerung von Simulationen aufgrund der partiellen Simulations-Ergebnisse.

Im TestBencher Pro ist Verilogger Pro – ein Verilog-Simulator – integriert. Auch externe Simulatoren wie z. B. Modelsim sind steuerbar.

TestBencher Pro Design flow



Vorteile der GUI

Als Profi erkennen Sie natürlich sofort, was hier geschrieben steht. Sie haben aber sicherlich auch schon komplexeren HDL Code gesehen.

TestBencher Pro erstellt aus einer Vielzahl von Timingdiagrammen, die unterschiedliche Transaktionen widerspiegeln, eine fertige Testbench für viele Simulatoren. Ihnen wird die Knochenarbeit des Code Hacking für Testbenches weitestgehend abgenommen.

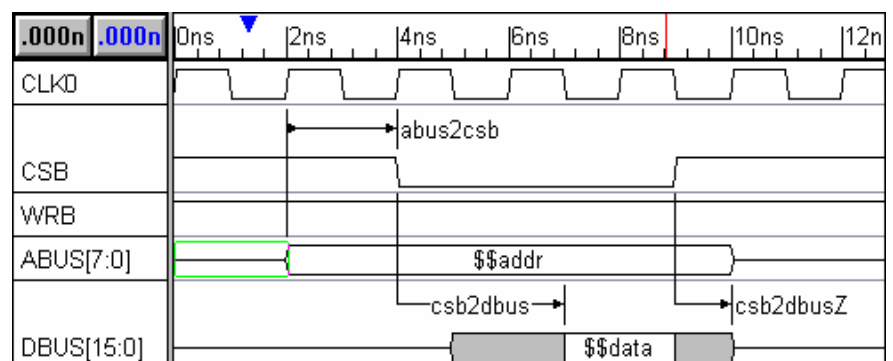
Den Inhalt der Timingdiagramme erkennen Sie auch noch sehr gut nach langer Zeit, nach vielen Änderungen und auch wenn das Diagramm vor langer Zeit vom Kollegen entwickelt wurde.

Die Timingdiagramme sind wieder einsetzbar, in mehreren Projekten einsetzbar und sprachunabhängig. Die Diagramme sind also in mehreren Projekten, auch mit unterschiedlicher HDL einsetzbar.

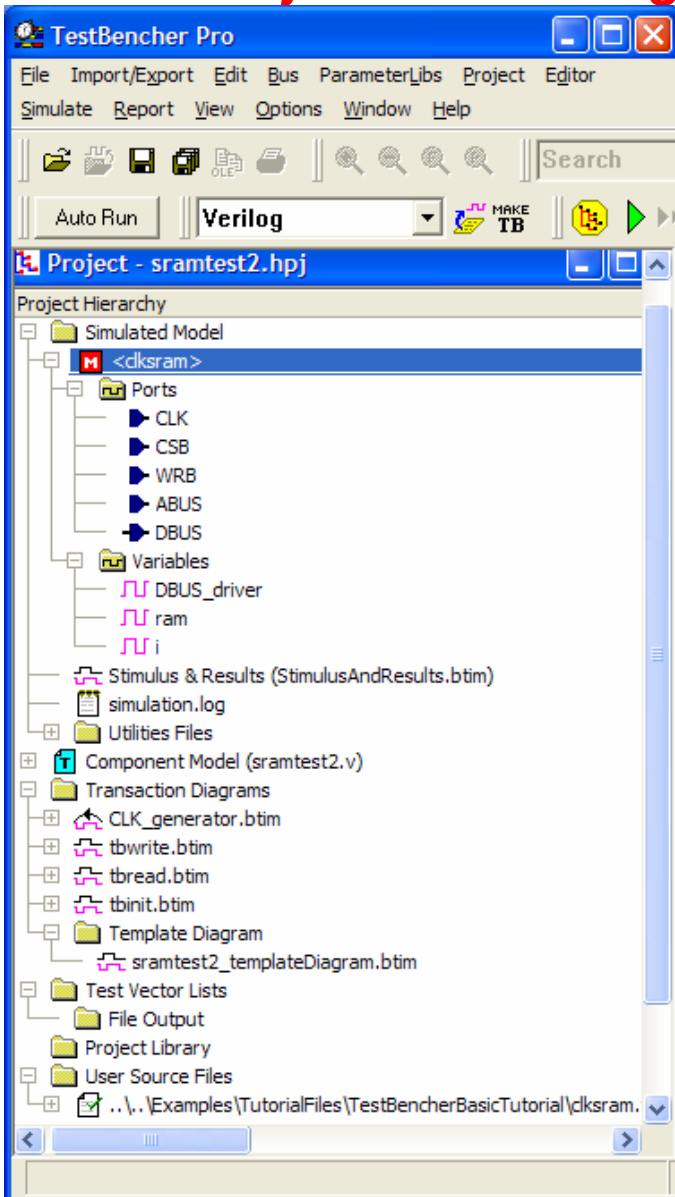
```

module testbench;
...
task write(addr,data,csb2dbus);
input [7:0] addr;
input [15:0] data;
input [1:0] csb2dbus;
begin
    ABUS = addr;
    @(posedge CLK0) //required abus2csb setup
    CSB = 1'b0;
    repeat (csb2dbus) @CLK0;
    DBUS = data;
    @(posedge CLK0)
    CSB = 1'b1;
    DBUS = 'hz;
    ABUS = 'hz;
end
endtask
...
Endmodule

```



Mit der Projekt-Verwaltung von TestBencher Pro



haben Sie alles übersichtlich im Griff.

Im Ordner „User Source Files“ befinden sich die zu testenden Source Files. TestBencher Pro extrahiert aus den Source Codes per Knopfdruck alle für die Simulation wichtigen Informationen und stellt das MUT (Model Under Test) mit den Ports und Variablen in den Ordner „Simulated Model“.

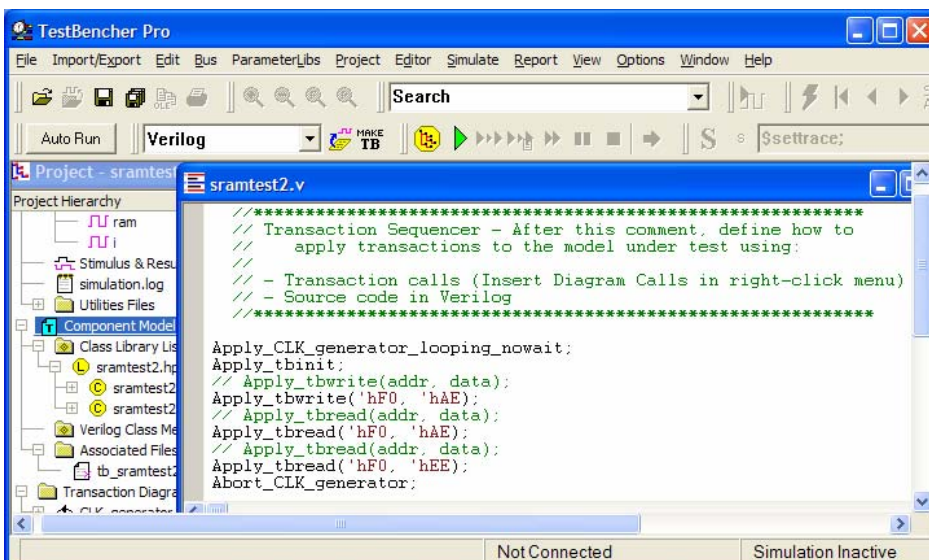
Damit der Anwender bei der Entwicklung der Transaktions-Diagramme schnell voran kommt, stellt TestBencher Pro in den Ordner „Template Diagram“ ein Timing Diagramm, das bereits alle erforderlichen Signale enthält.

Mit dem grafischen Timing Editor aus TimingDiagrammer Pro sind mit dem Template als Basis schnell alle Transaktionen spezifiziert. Die Transaktions-Diagramme sind im Ordner „Transaction Diagrams“ zu sehen.

Die Timing Diagramme sind natürlich auch in Dokumentationen verwendbar.

Nachdem ein Transaktions-Diagramm fertig ist, wird es gespeichert – TestBencher Pro erstellt im Hintergrund sofort unmerklich den Source Code für die Testbench.

Mit TestBencher Pro bestimmen Sie den Ablauf der Simulation



Im Transaction Sequencer wird der Ablauf der Simulation mit der Reihenfolge der Transaktions Diagramme definiert. TestBencher Pro stellt dem Anwender auch für diesen Vorgang eine GUI zur Verfügung.

Daten werden hier ebenfalls übergeben. Die Daten der Lesezyklen enthalten die erwarteten Werte. In diesem Beispiel erwartet ein Lesezyklus zur Demonstration falsche Werte.

TestBencher Pro

erstellt Ihnen danach die komplette Testbench – natürlich auch umfangreichere und komplexere Testbenches.

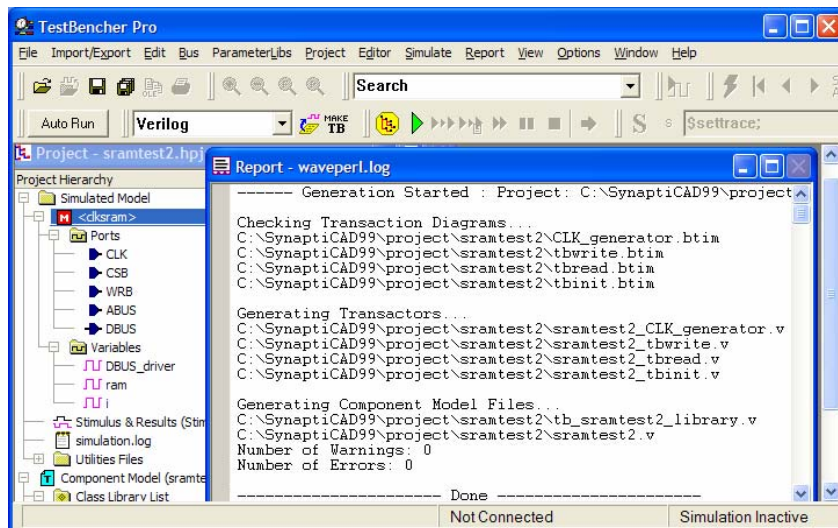
```

module clk_sram(CLK,CSB,WRB,ABUS,DBUS);
input CLK;
input CSB;
input WRB;
input [7:0] ABUS;
inout [7:0] DBUS;

reg [7:0] DBUS_driver;
wire [7:0] DBUS = DBUS_driver;
reg [7:0] ram[0:255];

integer i;
initial
begin
    //initialize all RAM cells to 0 at startup
    DBUS_driver = 8'bzzzzzzzz;
    for (i=0; i < 4095; i = i + 1)
        ram[i] = 0;
    end

always @(negedge CLK)
begin
    if (CSB == 1'b0)
    if (WRB == 1'b0)
    begin
        ram[ABUS] = DBUS;
        $display("In clk_sram at %t: Writing %h to address %h", $time, ram[ABUS], ABUS);
    end
    else
    begin
        DBUS_driver = ram[ABUS];
        $display("In clk_sram at %t: Reading %h to address %h", $time, ram[ABUS], ABUS);
    end
    else
        DBUS_driver = 8'bzzzzzzzz;
    end
end
endmodule
    
```



Damit alles korrekt läuft hat TestBencher Pro viele Protokolle. Hervorzuheben ist hier das Protokoll nach der automatischen Entwicklung der Testbench – in diesem Beispiel ein gutes Ergebnis ohne Warnings und Fehler.

TestBencher Pro

bietet das Komfort-Programm für viele Simulatoren – für mehr als hier angezeigt. Die Simulationsergebnisse werden über VCD Dateien in den TestBencher Pro zurückgeführt. TestBencher Pro läuft auf Windows, Linux und Solaris.

