

論理回路入門（3）

Quartus II と ModelSim の使い方

李 亜民

2022 年 10 月 11 日 (火)

Quartus II と ModelSim の使い方

ポイント

- Quartus II の使い方
 - ▶ 回路図で回路の作成
 - ▶ 回路解析と論理合成
 - ▶ 回路図から Verilog HDL ファイルに変換
- ModelSim の使い方
 - ▶ シミュレーション用のテストベンチの作成
 - ▶ Verilog HDL ファイルのコンパイル
 - ▶ テストベンチを使用してシミュレーション
 - ▶ 波形の確認

回路設計用ツール

● Quartus II

- ▶ Intel・アルテラの FPGA、SoC FPGA、CPLD の開発に対して、エントリーから論理合成、デバイスのプログラミングや評価・検証までの全ての開発フローをカバーする開発ツールである。
- ▶ Quartus II Web Edition は、無償パッケージであり、主要な FPGA および CPLD デザインをサポートするエントリーレベル・バージョンのデザイン・ソフトウェアである。ライセンス申請が不要なので、ソフトウェアをダウンロードしてインストール後に直ぐにお使い頂ける。
- ▶ Quartus II Subscription Edition は有償である。

動作検証用ツール

● ModelSim

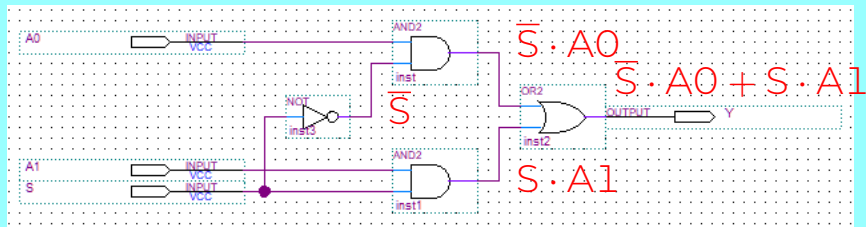
- ▶ もともと Mentor Graphics 社によって開発された。
- ▶ Quartus II v9.1 までは波形シミュレータが内蔵されていました。しかし、HDL を使用した設計が標準的となってきた事で、Quartus II v10.0 以降は波形シミュレータに代わり、ModelSim-Altera Edition が用意された。
- ▶ ModelSim-Altera は、Intel・アルテラ社 FPGA / CPLD の動作検証を行うための HDL シミュレータである。
- ▶ ModelSim-Altera Starter Edition ソフトウェアを無償で使用可能（ModelSim-Altera Edition は有償）である。

回路設計の例

論理式の例

$$Y = \bar{S} \cdot A0 + S \cdot A1$$

その論理式の論理回路



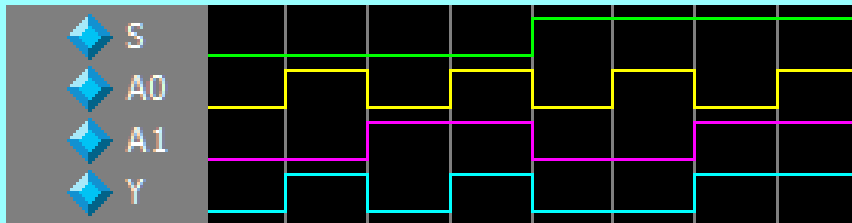
優先順位: (高い) 否定 → 論理積 → 論理和 (低い)

動作検証の例

論理式の例

$$Y = \bar{S} \cdot A0 + S \cdot A1$$

論理回路の動作検証の波形



次に、その回路設計と動作検証の方法を説明する。

Quartus II と ModelSim の使い方

- 1 Quartus II 13.1 を実行する前に、作業フォルダーを作成する。
例えば、ドキュメント \logic (全角文字は使用しないこと)
- 2 回路設計と動作検証シミュレーション
 - 1 Quartus II 13.1 を起動する。
 - 1 作成した作業フォルダーにプロジェクトを作成する。
 - 2 回路図で回路を作成する。
 - 3 回路解析と論理合成。エラーがある場合は回路を修正する。
 - 4 回路図から Verilog HDL ファイルに変換する。
 - 5 シミュレーション用の Verilog HDL テストベンチを作成する。
 - 2 ModelSim-Quartus 10.1d を起動する。
 - 1 作業フォルダーに移動する。
 - 2 Verilog HDL ファイルをコンパイルする。
 - 3 テストベンチを使用してシミュレーションする。
 - 4 波形を確認する。エラーがある場合は回路を修正する。
 - 3 Quartus II では、後で使用するために、回路のシンボルを作成する。

1. 作業フォルダーを作成する

まず最初に、作業フォルダーを作成する。例えば

C:\Users\your_account\Documents\logic

これからのすべてのデザインはこの作業フォルダーで行われなければならない。



2. Quartus II 13.1 を起動する

Quartus II 13.1 を起動する。
一回目だけ

Run the Quartus II software を選択する。
OK をクリックする。
Home window を閉じる。

3. 新規プロジェクト — mux2x1

What is the working directory for this project?
C:/Users/your_account/Documents/logic

What is the name of this project?
mux2x1

What is the name of the top-level entity for this project? The name is case-sensitive and must exactly match the entity name in the design file.
mux2x1

Use Existing Project Settings...

< Back Next > Finish Cancel Help

File ▶ New Project Wizard...
Next> をクリックする。
作業フォルダーとプロジェクト名を入力する。
Working directory:
C:/Users/your_account/Documents/logic
Project name: mux2x1
Finish をクリックする。

4. 回路編集ウィンドウを開く

回路編集ウィンドウを開く。

File ▶ New

Block Diagram/Schematic File を選択する。

OK をクリックする。

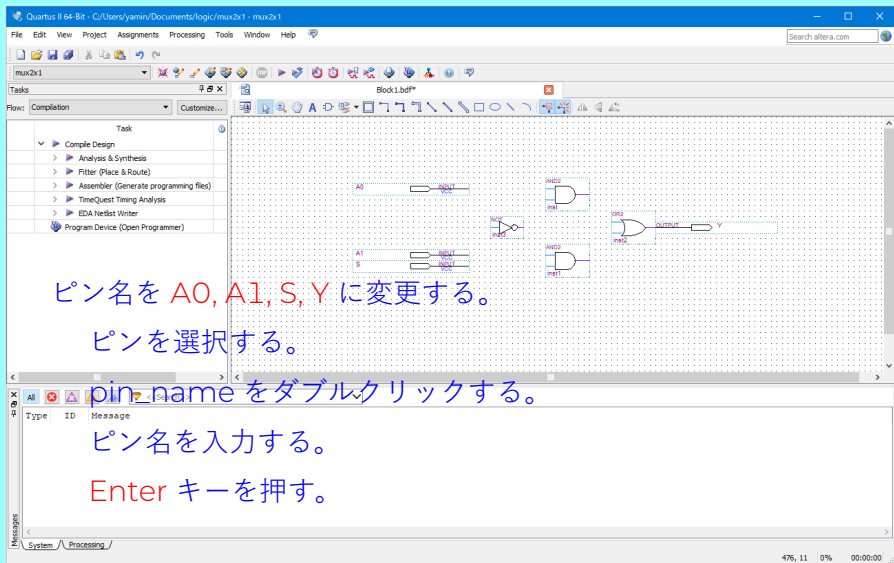
5. 回路コンポーネントを入力する

回路コンポーネントを入力する。
編集ウィンドウ内をダブルクリックする。
Nameフィールドに and2 を入力する。
OK をクリックする。

6. 論理ゲートとピンを入力する

or2, not, input, output などのコンポーネントを入力する。
Ctrl キーを押しながらマウスでコンポーネントをコピーできる。

7. ピン名を割り当てる



The screenshot shows the Quartus II 64-bit software interface. The main window displays a logic diagram with several components: two AND gates (AND2), two OR gates (OR2), a NOT gate (NOT), and two output pins (A0, A1, S, Y). The pins are labeled with their respective names. The message window at the bottom shows a list of messages with columns for Type, ID, and Message.

ピン名を A0, A1, S, Y に変更する。

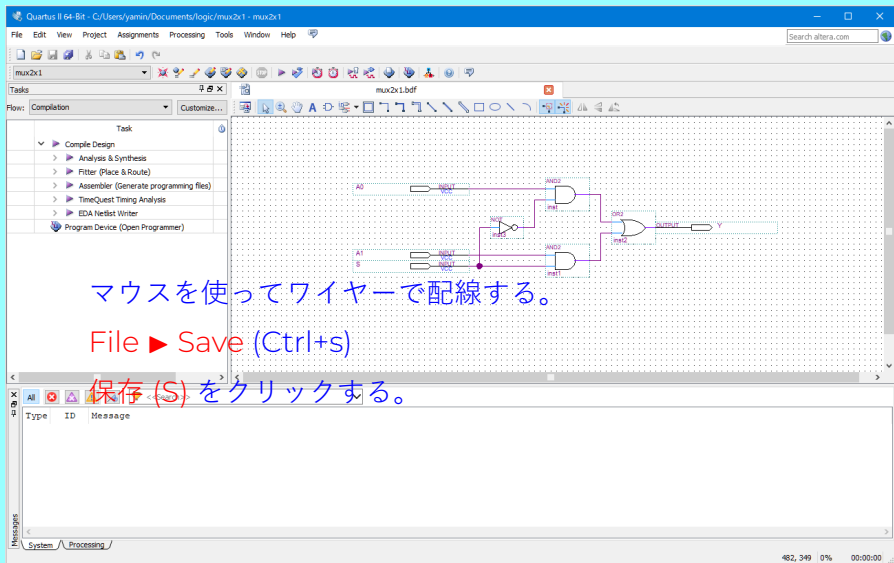
ピンを選択する。

pin_name をダブルクリックする。

ピン名を入力する。

Enter キーを押す。

8. 配線と保存する



マウスを使ってワイヤーで配線する。

File ▶ Save (Ctrl+s)

保存 (S) をクリックする。

9. 回路解析と論理合成

The screenshot shows the Quartus II 64-Bit software interface. The main window displays the 'Flow Summary' for the project 'mux2x1.bdf'. The flow is currently in the 'Analysis & Synthesis' step, which is highlighted in blue. The summary table shows the following statistics:

| Category | Count |
|------------------------------------|--|
| Flow Status | Successful - Tue Apr 24 14:29:44 2018 |
| Quartus II 64-Bit Version | 13.1.0 Build 162 10/23/2013 S1 Web Edition |
| Revision Name | mux2x1 |
| Top-level Entity Name | mux2x1 |
| Family | Cyclone IV GX |
| Total logic elements | 1 |
| Total combinational functions | 1 |
| Dedicated logic registers | 0 |
| Total registers | 0 |
| Total pins | 4 |
| Total virtual pins | 0 |
| Total memory bits | 0 |
| Embedded Multiplier 9-bit elements | 0 |
| Total GXB Receiver Channel PCS | 0 |
| Total GXB Receiver Channel PMA | 0 |
| Total GXB Transmitter Channel PCS | 0 |
| Total GXB Transmitter Channel PMA | 0 |
| Total PLLs | 0 |

Below the flow summary, a message window is open, showing the following messages:

```
Command: quartus_map --read_settings_files=on --write_settings_files=off mux2x1 -o mux2x1
20028 Parallel compilation is not licensed and has been disabled
> 12021 Found 1 design units, including 1 entities, in source file mux2x1.bdf
12127 Elaborating entity "mux2x1" for the top level hierarchy
286030 Timing-Driven Synthesis is running
> 16010 Generating hard_block partition "hard_block:auto_generated_inst"
21057 Implemented 5 device resources after synthesis - the final resource count might be different
> Quartus II 64-Bit Analysis & Synthesis was successful. 0 errors, 1 warning
```

Overlaid on the screenshot is the text: **Processing Start ▶ Start Analysis & Synthesis (Ctrl+k)** and **エラーがある場合は回路を修正する。**

10. Verilog HDL ファイルに変換する

回路図から Verilog HDL ファイルに変換する。
回路編集ウィンドウ mux2x1.bdf を選択する。

File ▶ Create/Update
Create HDL Design File from Current File...
Verilog HDL を選択する。
OK をクリックする。

556, 9 100% 00:00:17

11. テストベンチを作成する

テキスト編集ウィンドウを開く。

File ▶ New

Verilog HDL File を選択する。

OK をクリックする。

```
Run: Quartus II 64-bit Create Verilog File
Command: quartus_map --read_settings_files=on --write_settings_files=off mux2x1 -c mux2x1 --convert_bdf_to_verilog=C:/Users/yamin/Documents/logic/
12021 Found 1 design units, including 1 entities, in source file mux2x1.bdf
12127 Elaborating entity "" for the top level hierarchy
Quartus II 64-bit Create Verilog File was successful. 0 errors, 0 warnings
```

12. テストベンチコード

ここにコードを入力する。
(シミュレーション用)
mux2x1_tb.v 名前を付けて保存する。
ModelSim で編集することもできる。

```
1 timescale 1ns/1ps // unit = 1 ns; accuracy = 1 ps
2 module mux2x1_tb; // test bench, no input / output pins
3   reg S,A0,A1; // reg type for inputs of mux2x1
4   wire Y; // wire type for outputs of mux2x1
5   mux2x1 il (.A0(A0), .A1(A1), .S(S), .Y(Y)); // invoke mux2x1
6   initial begin // input signal patterns          S A1 A0
7     #0 S = 0; A1 = 0; A0 = 0; //          0 ns 0 0 0
8     #1 S = 0; A1 = 0; A0 = 1; // 0 + 1 = 1 ns 0 0 1
9     #1 S = 0; A1 = 1; A0 = 0; // 1 + 1 = 2 ns 0 1 0
10    #1 S = 0; A1 = 1; A0 = 1; // 2 + 1 = 3 ns 0 1 1
11    #1 S = 1; A1 = 0; A0 = 0; // 3 + 1 = 4 ns 1 0 0
12    #1 S = 1; A1 = 0; A0 = 1; // 4 + 1 = 5 ns 1 0 1
13    #1 S = 1; A1 = 1; A0 = 0; // 5 + 1 = 6 ns 1 1 0
14    #1 S = 1; A1 = 1; A0 = 1; // 6 + 1 = 7 ns 1 1 1
15    #1 $stop; // stop simulation after 8 ns
16   end
17 endmodule
18
```

| Type | ID | Message |
|-------------|-------|--|
| Information | | Running Quartus II 64-Bit Create Verilog File |
| Information | | Command: quartus_map --read_settings_files=on --write_settings_files=off mux2x1 -c mux2x1 --convert_bdf_to_verilog=C:/Users/yamin/Documents/logic/mux2x1.bdf |
| Information | 12021 | Found 1 design units, including 1 entities, in source file mux2x1.bdf |
| Information | 12127 | Elaborating entity "" for the top level hierarchy |
| Information | | Quartus II 64-Bit Create Verilog File was successful. 0 errors, 0 warnings |

13. ModelSim を起動

The screenshot shows the ModelSim ALTERA STARTER EDITION 10.1d interface. The main window displays a file tree with columns for Name, Type, and Path. The path for the selected file is `C:\Users\your_account\Documents\logic`. The Transcript window at the bottom shows the following commands:

```
# Reading C:/altera/13.1/modelsim_ase/tcl/vsim/pref.tcl
cd C:/Users/yamin/Documents/logic
ModelSim>
```

Red text annotations are overlaid on the image:

- At the top: **重要: 作業フォルダーに移動する:**
- Over the file tree: `C:\Users\your_account\Documents\logic`
- Over the Transcript window: **File ▶ Change Directory... to logic**

Blue text annotations are also present:

- Over the Transcript window: **ModelSim-Online 10.1d を起動する。**
- Over the Transcript window: **Welcomeのメニュー Close をクリックする。**
- Over the Transcript window: **作業フォルダーに移動する。**

15. “work” ディレクトリを作成する

ModelSim ALTERA STARTER EDITION 10.1d - Custom Altera Version

File Edit View Compile Simulate Add Library Tools Layout Bookmarks Window Help

Library: work

Compile Source Files

Library: work

ファイルの場所 (I): logic

db
incremental_db
output_files
mux2x1.v
mux2x1_tb.v

クイック アクセス
デスクトップ
ライブラリ

Create Library

Yes をクリックする (“work” ディレクトリの作成する。)

Done をクリックする (コンパイルウィンドウを閉じる。)

ファイル名(N): "mux2x1_tb.v" "mux2x1.v"

ファイルの種類(T): HDL Files (*.v;*.vhdl;*.vho;*.hdl;*.vov;*.vp)

Compile Done

Compile selected files together Default Options... Edit Source

Transcript

```
# Reading C:/altera/13.1/modelsim_ase/tcl/vsim/pref.tcl  
cd C:/Users/yamin/Documents/logic  
ModelSim>
```

16. シミュレーション

The screenshot shows the ModelSim ALTERA STARTER EDITION 10.1d interface. The main window displays a library of components. A "Start Simulation" dialog box is open, showing the "Design" tab with a tree view of the project files. The file "mux2x1_tb" is selected. The "Design Unit(s)" field is empty, and the "Resolution" is set to "default". The "Optimization" section has "Enable optimization" checked. The "OK" button is highlighted.

Transcript window content:

```
vlog -reportprogress 300 -work work C:/Users/yamin/Documents/logic/mux2x1.v
# Model Technology ModelSim ALTERA vlog 10.1d Compiler 2012.11 Nov  3 2012
# -- Compiling module mux2x1
#
# Top level modules:
#   mux2x1
vlog -reportprogress 300 -work work C:/Users/yamin/Documents/logic/mux2x1_tb.v
# Model Technology ModelSim ALTERA vlog 10.1d Compiler 2012.11 Nov  3 2012
# -- Compiling module mux2x1_tb
#
# Top level modules:
#   mux2x1_tb
ModelSim>
```

シミュレーション

Simulate or Simulation...

work.mux2x1_tb を選択する。

OK をクリックする。

17. シミュレーション時間を変更する

The screenshot shows the ModelSim ALTEA Starter Edition 10.1d interface. The top menu bar includes File, Edit, View, Compile, Simulate, Add, Process, Tools, Layout, Bookmarks, Window, and Help. The main workspace is divided into several panes:

- Design Unit:** Shows the hierarchy of the design, including #vsim_capacity#, mux2x1_tb, #INITIAL#6, and #I1.
- Objects:** Lists objects such as S, A0, A1, and Y.
- Processes (Active):** Lists active processes like #I1/#ASSIGN#39, #I1/#ASSIGN#41, #I1/#ASSIGN#43, #I1/#ASSIGN#45, and mux2x1_tb/#INITI...
- Wave - Default:** A timing diagram showing a signal named 'Msig' over time. The time axis ranges from 0 ns to 800 ns. A cursor is positioned at 100 ns, and the current time is 0.00 ns.
- Transcript:** Shows the command-line output of the simulation. The text in the transcript is as follows:

```
# Top level module:
# mux2x1
vlog -reportprogress 300 -work work C:/Users/yahin/Documents/logic/mux2x1_tb.v
# Model Technology ModelSim ALTEA vlog 10.1d Compiler 2012.11 Nov 2 2012
# -- Compiling module mux2x1_tb
# Top level module:
# mux2x1_tb
ModelSim> vsim -gui work.mux2x1_tb
# vsim -gui work.mux2x1_tb
# Loading work.mux2x1_tb
# Loading work.mux2x1_tb
VSDM 6>
```

Overlaid on the transcript pane are blue annotations:

- シミュレーション時間を 100 ns に変更 (しなくてもよい)
- Transcript が開いていない場合は開く。
- View ▶ Transcript

At the bottom of the window, the status bar shows 'Now: 0 ps Delta: 0' and 'sim::mux2x1_tb'.

18. 波形ウィンドウに信号を追加する

The screenshot displays the ModelSim ALTERA STARTER EDITION 10.1d interface. The main window shows the Verilog code for a testbench and a 2-to-1 multiplexer module. The transcript window at the bottom shows the following commands and output:

```
Model Technology ModelSim ALTERA vlog 10.1d Compiler 2013.11 Nov  4 2013
# -- Compiling module mux2x1_tb
#
# Top level modules:
#
#   mux2x1_tb
ModelSim> vsim -gui -work mux2x1_tb
# vsim -gui work/mux2x1_tb
# Loading work_mux2x1_tb
# Loading work_mux2x1
VSIM6> add wave .*
VSIM7> run
# Break in Module mux2x1_tb at C:/Users/yamin/Documents/logic/mux2x1_tb.v line 15
VSIM8>|
```

Annotations in the image:

- トランスクリプトウィンドウに「add wave /*」と入力する。
- トランスクリプトウィンドウに「run 100ns」と入力する。

19. 波形を確認する

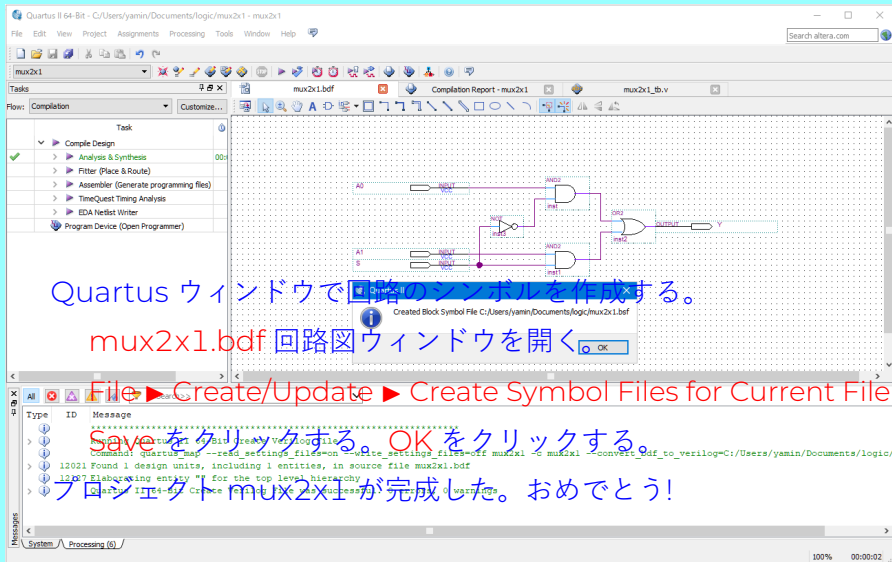
ここでマウスの右クリックで波形の表示を変更できる。
この Wave ウィンドウを Undock すると、File-Export-Image で波形を保存できる。

レポートに見やすい波形を掲載 (P6 を参照)

波形ウィンドウでは、波形の表示を変更できる。
波形から、回路の正しさを確認できる:

```
if (S == 0) Y = A0;  
else      Y = A1;
```

20. シンボルを作成する



Quartus II 64-Bit - C:/Users/yamin/Documents/logic/mux2x1 - mux2x1

File Edit View Project Assignments Processing Tools Window Help

Search altera.com

Tasks

Flow: Completion

Task

- Complete Design
 - Analysis & Synthesis
 - Fitter (Place & Route)
 - Assembler (Generate programming files)
 - TimeQuest Timing Analysis
 - EDA Netlist Writer
 - Program Device (Open Programmer)

mux2x1.bdf

Created Block Symbol File C:/Users/yamin/Documents/logic/mux2x1.bsf

File ▶ Create/Update ▶ Create Symbol Files for Current File

Save をクリックする。OK をクリックする。

プロジェクト mux2x1 が完成した。おめでとう!

| Type | ID | Message |
|-------------|-------|---|
| Information | 12021 | Found 1 design units, including 1 entities, in source file mux2x1.bdf |
| Information | 12027 | Elaborating entity "" for the top level hierarchy |
| Information | 12028 | Quartus II 64-Bit Create Symbol Files for Current File was successful. 0 warnings |

21. 新規プロジェクト — mux2x8

Quartus II 64-Bit

File Edit View Project Assignments Processing Tools Window Help

Search altera.com

Tasks

Flow: Completion

Task

- Complete Design
 - Analysis & Synthesis
 - Fitter (Place & Route)
 - Assembler (Generate program)
 - TimeQuest Timing Analysis
 - EDA Netlist Writer
- Program Device (Open Programm...)

New Project Wizard

Directory, Name, Top-Level Entity [page 1 of 5]

What is the working directory for this project?
C:\Users\yamin\Documents\logic

What is the name of this project?
mux2x8

What is the name of the top-level design entity for this project? This name is case sensitive and must exactly match the entity name in the design file.
mux2x8

Use Existing Project Settings...

Quartus II

Do you want to select a different project directory?
Projects in C:\Users\yamin\Documents\logic already exist as a project. If the projects share the same files, edits made while working on one project could affect the other project. Do you want to select a different project directory?

Yes No

Buy Software

View Quartus II Information

Documentation

Notification Center

Do you want to close current project "mux2x1"? Yes

プロジェクト名 mux2x8 を入力して、Finish をクリックする。

Project... Do you want to select a different project directory?
No をクリックする (同じ作業フォルダーを使用する)。

System Processing

756, 4 100% 00:00:02

22. 回路コンポーネントを入力する

The screenshot displays the Quartus II 64-bit IDE interface. The main workspace shows a schematic diagram with a grid background. A "Symbol" dialog box is open, allowing the user to select a component from a library. The "Libraries" pane on the left shows the project hierarchy, with "mux2x1" selected under the "Project" folder. The "Name:" field contains "mux2x1". The "Repeat-insert mode", "Insert symbol as block", and "Launch MegaWizard Plug-In" options are unchecked. The "MegaWizard Plug-In Manager..." button is visible. The schematic diagram in the background shows a component symbol with inputs labeled "A0", "A1", and "S", and an output labeled "Y".

Quartus II 64-Bit - C:/Users/yamin/Documents/logic/mux2x8 - mux2x8

File Edit View Project Assignments Processing Tools Window Help

Search altera.com

mux2x8

Tasks

Flow: Completion

Task

- Complete Design
 - Analysis & Synthesis
 - Fitter (Place & Route)
 - Assembler (Generate programming files)
 - TimeQuest Timing Analysis
 - EDA Netlist Writer
 - Program Device (Open Programmer)

Block2.bdf

Symbol

Libraries:

- Project
 - mux2x1
 - c:/altera/13.1/quartus/libraries/

Name: mux2x1

Repeat-insert mode

Insert symbol as block

Launch MegaWizard Plug-In

MegaWizard Plug-In Manager...

OK Cancel

Messages

System / Processing /

294,146 100% 00:00:02

23. マウスを使ってワイヤーで配線

A0[7..0]: 8ビット幅である。
A1[7..0]: 8ビット幅である。
Y[7..0]: 8ビット幅である。
注意: 二つの点を使うこと。

24. 回路解析と論理合成

The screenshot displays the Quartus II 64-Bit software interface. The main window shows the 'Flow Summary' for the project 'mux2x8.bdf'. The flow is completed successfully on Tuesday, April 24, 2018, at 18:12:49. The summary lists various design metrics:

| Item | Value |
|------------------------------------|--|
| Quartus II 64-Bit Version | 13.1.0 Build 162 10/23/2013 S1 Web Edition |
| Revision Name | mux2x8 |
| Top-level Entity Name | mux2x8 |
| Family | Cyclone IV GX |
| Total logic elements | 8 |
| Total combinational functions | 8 |
| Dedicated logic registers | 0 |
| Total registers | 0 |
| Total pins | 25 |
| Total virtual pins | 0 |
| Total memory bits | 0 |
| Embedded Multiplier 9-bit elements | 0 |
| Total GXB Receiver Channel PCS | 0 |
| Total GXB Receiver Channel PMA | 0 |
| Total GXB Transmitter Channel PCS | 0 |
| Total GXB Transmitter Channel PMA | 0 |
| Total PLLs | 0 |

The left-hand 'Tasks' pane shows the compilation flow: Compile Design (Analysis & Synthesis, Fitter, Assembler, TimeQuest, EDA Netlist Writer) and Program Device (Open Programmer). The status bar at the bottom indicates 495,355 resources, 100% completion, and a time of 00:00:03.

25. Verilog HDL ファイルに変換する

The screenshot displays the Quartus II 64-bit interface for a project named 'mux2x8'. The main workspace shows a logic diagram consisting of four 2-to-1 multiplexers (mux2x1) connected in a chain. The inputs are labeled A0, A1, and S, and the outputs are Y0, Y1, Y2, and Y3. A final output is labeled 'output' and 'Y[7:0]'. A dialog box titled 'Create HDL Design File for Current File' is open in the foreground, showing the file name 'rs/yamin/Documents/logic/mux2x8.v' and the file type set to 'Verilog HDL'. The dialog also includes an 'Add VHDL Statements...' button and 'OK', 'Cancel', and 'Help' buttons.

Tasks

- Complete Design
 - Analysis & Synthesis
 - Fitter (Place & Route)
 - Assembler (Generate programming files)
 - TimeQuest Timing Analysis
 - EDA Netlist Writer
 - Program Device (Open Programmer)

File name: rs/yamin/Documents/logic/mux2x8.v

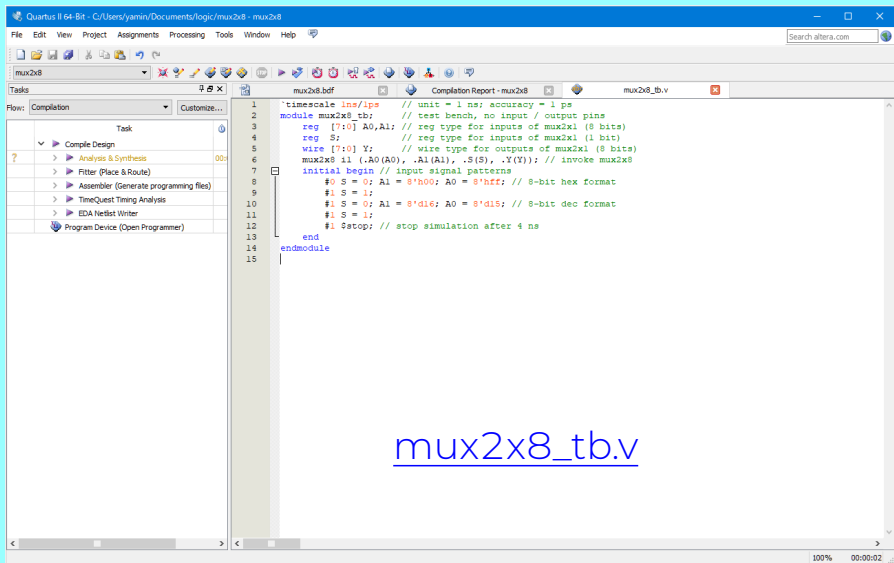
File type

- VHDL
- Verilog HDL

Add VHDL Statements...

OK Cancel Help

26. テストベンチを作成する



The screenshot shows the Quartus II 64-bit IDE interface. The main window displays a Verilog testbench named `mux2x8_tb.v` for a module named `mux2x8`. The code is as follows:

```
1 timescale 1ns/1ps // unit = 1 ns; accuracy = 1 ps
2 module mux2x8_tb; // test bench, no input / output pins
3   reg [7:0] A0,A1; // reg type for inputs of mux2x1 (8 bits)
4   reg S; // reg type for inputs of mux2x1 (1 bit)
5   wire [7:0] Y; // wire type for outputs of mux2x1 (8 bits)
6   mux2x8 il (.A0(A0), .A1(A1), .S(S), .Y(Y)); // invoke mux2x8
7   initial begin // input signal patterns
8     #0 S = 0; A1 = 8'h00; A0 = 8'hff; // 8-bit hex format
9     #1 S = 1;
10    #1 S = 0; A1 = 8'd16; A0 = 8'd15; // 8-bit dec format
11    #1 S = 1;
12    #1 $stop; // stop simulation after 4 ns
13  end
14 endmodule
15
```

The left sidebar shows the 'Tasks' pane with the 'Complete Design' task selected. The status bar at the bottom indicates 100% zoom and a simulation time of 00:00:02.

[mux2x8_tb.v](#)

27. シミュレーション

The screenshot displays the ModelSim ALTEA STARTER EDITION 10.1d interface. The main window shows a simulation of a mux2x8_tb module. The design tree on the left lists the module and its components, including #INITIAL#7. The wave window on the right shows the signals A0, A1, S, and Y over time, with a time scale of 100 ns. The transcript window at the bottom shows the simulation commands and output, including the compilation of the module and the execution of the vsim command.

```
# -- Compiling module mux2x8_tb
#
# Top level modules:
#   mux2x8_tb
VSIM 10> vsim -gui work.mux2x8_tb
# Loading work.mux2x8_tb
# Loading work.mux2x8
# Loading work.mux2x1
VSIM 11> add wave .*
VSIM 12> run
# Break in Module mux2x8_tb at C:/Users/yamin/Documents/logic/mux2x8_tb.v line 12
VSIM 13>
```

Now: 4 ns Delta: 0 sim:mux2x8_tb.#INITIAL#7 0 ps to 4 ns

28. ブロックシンボルを作成

The screenshot shows the Quartus II IDE interface. The main workspace displays a schematic diagram for a 2-to-1 multiplexer implementation using eight 'mux2x1' blocks. The blocks are arranged in two columns of four, connected in a hierarchical tree structure. The left column blocks have inputs A0, A1, and S, and outputs Y0 through Y3. The right column blocks have inputs A0, A1, and S, and outputs Y4 through Y7. The final output is connected to a signal named 'output' with value 'Y[7:0]'. A dialog box titled 'Quartus II' is open in the foreground, displaying the message: 'Created Block Symbol File C:/Users/yamin/Documents/logic/mux2x8.bsf'. The dialog has an 'OK' button.

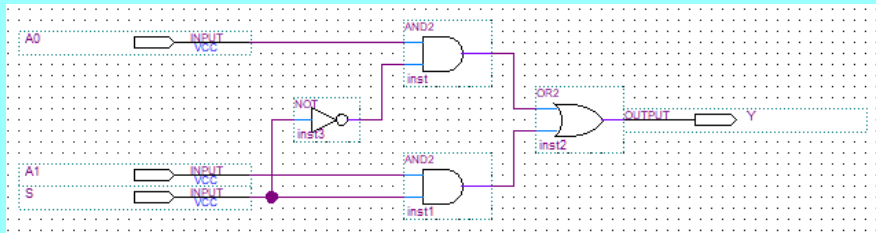
Quartus II と ModelSim の使い方

まとめ

- Quartus II の使い方
 - ▶ 回路図で回路の作成
 - ▶ 回路解析と論理合成
 - ▶ 回路図から Verilog HDL ファイルに変換
- ModelSim の使い方
 - ▶ シミュレーション用のテストベンチの作成
 - ▶ Verilog HDL ファイルのコンパイル
 - ▶ テストベンチを使用してシミュレーション
 - ▶ 波形の確認

課題 III (100 点)

次の回路を設計し動作検証シミュレーションして下さい。



プロジェクト名は `mux2x1` にすること。

テストベンチ [mux2x1_tb.v](#) を使って下さい。

レポートの必須項目: (1) 回路図、(2) 論理式、(3) 真理値表、(4) シミュレーション波形、(5) 波形の説明。

[GBC【論理回路入門チーム】論理回路入門レポートの例](#)

発展：NAND ゲートの CMOS 型実装

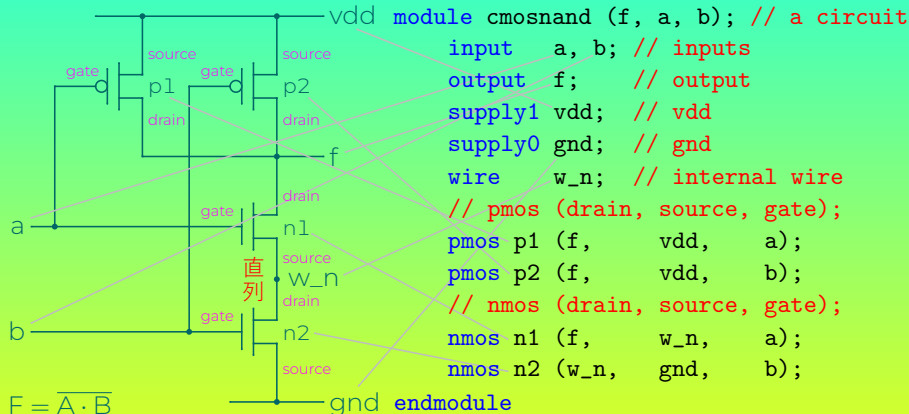
Quartus II は CMOS トランジスタ入力をサポートしていない

発展：Verilog HDL による NAND ゲートの CMOS 型実装

(回路図ではなく言語を使用して回路を実装する)

発展：NAND ゲートの CMOS 型実装

Verilog HDL による NAND ゲートの CMOS 型実装



↑ 青色：キーワード

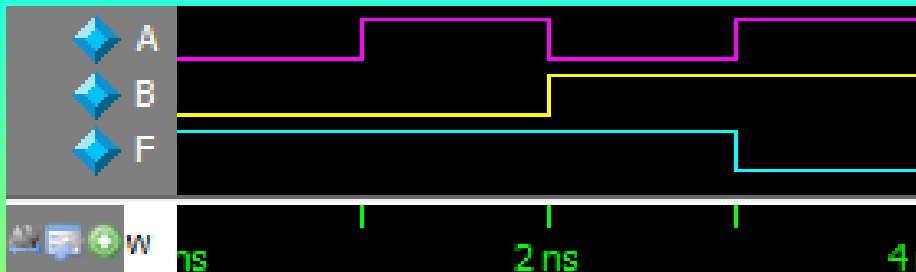
発展：NAND ゲートの CMOS 型実装

```
module cmosnand (f, a, b); // a circuit
    input  a, b; // inputs
    output f;    // output
    supply1 vdd; // vdd
    supply0 gnd; // gnd
    wire   w_n; // internal wire
    // pmos (drain, source, gate);
    pmos p1 (f,      vdd,  a);
    pmos p2 (f,      vdd,  b);
    // nmos (drain, source, gate);
    nmos n1 (f,      w_n,  a);
    nmos n2 (w_n,    gnd,  b);
endmodule
```


発展：NAND ゲートの CMOS 型実装

```
'timescale 1ns/1ns
module cmosnand_tb; // a test bench, not a circuit
    reg A, B;
    wire F;
    cmosnand not_and (F, A, B); // invoke cmosnand
    initial begin
        A = 0;
        B = 0;
        #4 $stop;
    end
    always #1 A = ~A;
    always #2 B = ~B;
endmodule
```

発展：NAND ゲートの CMOS 型実装



| A | B | $A \cdot B$ | $\overline{A \cdot B}$ |
|---|---|-------------|------------------------|
| 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

発展：自由練習

① 自由練習：P39 - P42 を参照して、PMOS と NMOS トランジスタを使用して、Verilog HDL で 次のゲートを設計とシミュレーションして下さい。

(1) $F = \overline{A + B}$

(2) $F = \overline{A \cdot B \cdot C}$ 課題 I 問題 2(1) の検証

(3) $F = \overline{A + B + C}$ 課題 I 問題 2(2) の検証

② 自由練習：以下の URL にある問題について、回路シミュレーションを行って答えを求めて下さい (ただし、出力信号名は Z1, Z2, Z3, Z4 にすること)。

http://www.fe-siken.com/kakomon/26_haru/q22.html