

# Contents

## 第一部 入門編

<b>第1章 やさしいVerilog HDL記述入門</b> .....	<b>14</b>
1.1 HDLって何だ? .....	14
1.1.1 HDL記述と論理合成	14
1.1.2 HDL設計のメリット/デメリット	16
1.1.3 記述レベル	18
1.2 加算回路のHDL記述 .....	19
1.2.1 加算演算子による加算回路	19
1.2.2 フル・アダー呼び出しによる加算回路1 ——ポート リスト順の接続	21
1.2.3 フル・アダー呼び出しによる加算回路2 ——名まえによる接続	24
1.2.4 セル呼び出しによる加算回路	25
1.3 カウンタのHDL記述 .....	26
1.3.1 加算演算子によるバイナリ カウンタ	26
1.3.2 1ビットのカウンタ ユニット呼び出しによるバイナリ・カウンタ	28
1.3.3 セル呼び出しによるバイナリ・カウンタ	29
1.4 シミュレーションしてみよう .....	29
1.4.1 加算回路のシミュレーション記述	29
1.4.2 加算回路のシミュレーション結果	33
1.4.3 カウンタのシミュレーション記述	34
1.4.4 カウンタのシミュレーション結果	35
1.5 論理合成してみよう .....	35
1.5.1 論理合成できるのはVerilog HDLの一部	36
1.5.2 加算回路の論理合成	37
1.5.3 カウンタの論理合成	38
<b>第2章 もう少し進んだVerilog HDL記述</b> .....	<b>40</b>
2.1 電子サイコロ .....	40
2.1.1 電子サイコロの仕様	40
2.1.2 電子サイコロ回路概要	40
2.1.3 電子サイコロのHDL記述	41
2.1.4 電子サイコロのシミュレーション記述と結果	44

<b>2.2 電子錠</b> .....	<b>46</b>
2.2.1 電子錠の仕様	46
2.2.2 電子錠の回路概要	47
2.2.3 電子錠のHDL記述	47
2.2.4 電子錠のシミュレーション記述と結果	52
<b>2.3 電子サイコロと電子錠の論理合成</b> .....	<b>53</b>

## 第二部 回路記述編

<b>第3章 文法概略と基本記述スタイル</b> .....	<b>56</b>
<b>3.1 文法を少々</b> .....	<b>56</b>
3.1.1 モジュール構造	56
3.1.2 論理値と数値表現	58
3.1.3 データ型	58
3.1.4 多ビット信号	60
3.1.5 演算子と演算優先順位	62
3.1.6 等号演算と関係演算	64
3.1.7 接続演算	64
3.1.8 リダクション演算	66
3.1.9 回路記述に必要な構文	67
<b>3.2 回路記述早わかり</b> .....	<b>69</b>
3.2.1 assign文による組み合わせ回路	69
3.2.2 functionによる組み合わせ回路	70
3.2.3 always文による順序回路	71
3.2.4 下位モジュール呼び出し	72
<b>第4章 組み合わせ回路のHDL記述</b> .....	<b>75</b>
<b>4.1 基本ゲート回路</b> .....	<b>75</b>
4.1.1 プリミティブ・ゲートを用いたゲート回路	75
4.1.2 論理式を用いたゲート回路	76
<b>4.2 セレクタ</b> .....	<b>77</b>
4.2.1 2 to 1セレクタ	77
4.2.2 4 to 1セレクタ	80
4.2.3 3 to 1セレクタ	83
<b>4.3 デコーダ</b> .....	<b>85</b>
4.3.1 等号演算によるデコーダ	85
4.3.2 if文によるデコーダ	85
4.3.3 case文によるデコーダ	86

<b>4.4 エンコーダ</b> .....	<b>87</b>
4.4.1 if文によるエンコーダ	87
4.4.2 casex文によるエンコーダ	88
4.4.3 for文によるエンコーダ	88
4.4.4 always文によるエンコーダ	91
<b>4.5 演算回路</b> .....	<b>92</b>
4.5.1 加算回路	92
4.5.2 減算回路	93
4.5.3 定数加算回路	95
4.5.4 バレル・シフタ	96
4.5.5 乗算回路	96
<b>4.6 比較回路</b> .....	<b>97</b>
<b>4.7 組み合わせ回路で作るROM</b> .....	<b>97</b>
<b>4.8 3ステート信号の記述</b> .....	<b>99</b>
<b>4.9 組み合わせ回路の論理合成</b> .....	<b>102</b>
<b>第5章 順序回路のHDL記述</b> .....	<b>103</b>
<b>5.1 非同期型フリップフロップ</b> .....	<b>103</b>
5.1.1 SRフリップフロップ	103
5.1.2 Dラッチ	104
5.1.3 Dフリップフロップ	105
5.1.4 非同期セット/リセット付きDフリップフロップ	105
5.1.5 非同期セット/リセット付きJKフリップフロップ	107
5.1.6 非同期リセット付きTフリップフロップ	108
<b>5.2 同期型フリップフロップ</b> .....	<b>108</b>
5.2.1 同期SRフリップフロップ	108
5.2.2 同期セット/リセット付きDフリップフロップ	109
5.2.3 同期セット/リセット付きJKフリップフロップ	110
5.2.4 同期リセット付きTフリップフロップ	111
5.2.5 同期設計用フリップフロップの構成	111
<b>5.3 各種カウンタ</b> .....	<b>113</b>
5.3.1 バイナリ・カウンタ	113
5.3.2 アップ/ダウン・カウンタ	113
5.3.3 グレイ・コード・カウンタ	114
5.3.4 手抜きグレイ・コード・カウンタ	116
5.3.5 ジョンソン・カウンタ	117
5.3.6 不正ループ対策版ジョンソン・カウンタ	118
5.3.7 リング・カウンタ	119
5.3.8 ディバイダ(分周回路)	120

5.4	シフト・レジスタ	122
5.4.1	シリアル・パラレル変換	122
5.4.2	シリアル・パラレル変換 NG編その1	123
5.4.3	シリアル・パラレル変換 NG編その2	123
5.4.4	究極のシリアル・パラレル変換	124
5.4.5	パラレル・シリアル変換	126
5.5	レジスタ・ファイル	127
5.6	ステート・マシン	128
5.6.1	ステート マシンの目的	128
5.6.2	ステート・マシンの構成法	130
5.6.3	ステート マシンの記述例	130
5.7	順序回路の論理合成	137

## 第三部 シミュレーション&応用編

第6章	シミュレーション・モデル	140
6.1	シミュレーション・モデルの必要性	140
6.1.1	論理回路シミュレーションの今昔	140
6.1.2	シミュレーション モデルとは	142
6.2	シミュレーション・モデルの記述例	142
6.2.1	ROMシミュレーション・モデル	143
6.2.2	RAMシミュレーション モデル	144
6.2.3	A-D変換シミュレーション・モデル	146
6.2.4	D-A変換シミュレーション・モデル	148
6.3	タスクによるシミュレーション・モデル	150
6.4	シミュレーション・モデルの使いかた	153
第7章	シミュレーション記述	155
7.1	シミュレーション記述概要	155
7.1.1	シミュレーションの入力記述	156
7.1.2	シミュレーションの出力記述	157
7.1.3	論理合成後のシミュレーション	159
7.1.4	シミュレーションの流れ	159
7.1.5	モジュール・アイテムとステートメント	160
7.1.6	initial文とalways文	161
7.2	ステートメント	163
7.2.1	for文	163
7.2.2	while文	163

7.2.3	repeat文	164
7.2.4	forever文	164
7.2.5	タイミング制御	164
7.2.6	wait文	165
7.2.7	イベント宣言とイベント起動	165
7.2.8	begin~endとfork~join	167
7.3	タスク	167
7.3.1	CPUによるRead/Writeモデル	167
7.3.2	各種タスク テクニック	169
7.4	システム・タスク	172
7.4.1	ファイル Read/Writeシステム・タスク	172
7.4.2	表示用システム・タスク	173
7.4.3	フォーマットされたファイル出力システム タスク	175
7.4.4	値を返すシステム・タスク	176
7.4.5	シミュレーションの実行を制御するシステム タスク	176
7.5	その他のシミュレーション記述	177
7.5.1	force文とrelease文	177
7.5.2	さまざまな遅延	177
7.5.3	階層アクセス	178
7.5.4	コンパイラ指示子	179
7.5.5	コメント	181
第8章	電子銃の拡張	183
8.1	回路仕様	183
8.2	クロック生成部	185
8.3	キー入力部	186
8.4	電子銃本体	190
8.5	表示部	198
8.6	回路検証	202
8.7	FPGAによる動作確認	210

## Appendix I Verilog HDL文法概要 216

I.1	Verilog HDL文法	216
I.1.1	ソース・テキスト	216
I.1.2	宣言	217
I.1.3	プリミティブ・インスタンス	218
I.1.4	モジュール インスタンス生成	219
I.1.5	UDP宣言とインスタンス生成	220

I.1.6	動作記述	220
I.1.7	specify記述	222
I.1.8	式	224
I.1.9	一般事項	226
I.1.10	予約語一覧	226
<b>I.2</b>	<b>文法要約</b>	<b>227</b>
I.2.1	モジュール構造	227
I.2.2	モジュール構成要素	227
I.2.3	ステートメント	231
I.2.4	式	234
I.2.5	識別子	235
<b>Appendix II</b>	<b>Verilog-2001</b>	<b>236</b>
<b>II.1</b>	<b>回路記述向きの拡張</b>	<b>236</b>
II.1.1	宣言の拡張	236
II.1.2	構文の拡張	238
II.1.3	配列や演算の拡張	242
<b>II.2</b>	<b>テストベンチ向きの拡張</b>	<b>245</b>
<b>II.3</b>	<b>コンパイル時の拡張</b>	<b>249</b>
コラムA	信号名の付けかた	23
コラムB	ライブラリって何種類ある？	31
コラムC	組み合わせ回路と順序回路	33
コラムD	always文は重要	65
コラムE	always文による組み合わせ回路	74
コラムF	functionかalwaysか	82
コラムG	代入記号の使い分け	90
コラムH	case文のdefaultの役割	100
コラムI	論理合成がうまくいかない例	117
コラムJ	ブロッキング代入文とノン ブロッキング代入文	124
コラムK	「HDL流」記述スタイル	136
コラムL	順序回路の初期化は非同期リセット	145
コラムM	specifyブロック	148
コラムN	乱数を入れちゃえ	170
コラムO	呪われたシミュレーション	178
コラムP	厳密より現実 ～セット/リセット付きD-FFは正確に表現できない～	201
参考文献		250
索引		251