

報道各位

2026 年 10 月 8 日
株式会社インプレス

Claude Code をフル活用! 自律型エージェントで設計効率を最大化せよ!

『AI 時代の FPGA 開発 Claude Code と LLM で加速する FPGA 設計入門』発行

技術の泉シリーズ 10 月の新刊

インプレスグループで IT・デザイン関連メディア事業を展開する株式会社インプレス(本社:東京都千代田区、代表取締役社長:高橋隆志)は、『AI 時代の FPGA 開発 Claude Code と LLM で加速する FPGA 設計入門』(著者:石原 ひでみ)をインプレス NextPublishing より発行いたします。

最新の知見を発信する『技術の泉シリーズ』は、「技術書典」や「技術書同人誌博覧会」をはじめとした各種即売会や、勉強会・LT 会などで頒布された技術同人誌を底本とした商業書籍を刊行し、技術同人誌の普及と発展に貢献することを目指します。

『AI 時代の FPGA 開発 Claude Code と LLM で加速する FPGA 設計入門』

<https://nextpublishing.jp/isbn/9784295605263>



著者:石原 ひでみ

小売希望価格:電子書籍版 1,800 円(税別)／印刷書籍版 2,000 円(税別)

電子書籍版フォーマット:EPUB3

印刷書籍版仕様:B5 版／カラー／本文 110 ページ

ISBN:978-4-295-60526-3

企画編集:合同会社技術の泉出版

発行:インプレス NextPublishing

発売:株式会社インプレス

発売日:2026 年 10 月 9 日

<<発行主旨>>

本書は、急速に進化する生成 AI 技術を FPGA 開発にどのように取り入れるべきか、その最前線を解説する実践的な一冊です。従来の開発フローを劇的に変える「エージェント・コーディング」に焦点を当て、Claude Code などの最新ツールを用いた自律的な設計手法を詳しく紹介しています。

特に注目すべきは、生成 AI が陥りやすい HDL コーディングの失敗例に関する解説です。BRAM 推論の失敗やブロッキング・ノンブロッキング代入の混在など、ハードウェア設計特有の罠を具体的に挙げ、AI を正しく導くためのプロンプト技術やデバッグ手法を伝授します。

さらに、マルチエージェントシステムによるチーム開発の事例や、SystemVerilog の功罪、そして AI 時代の設計思想についても深く考察しています。単なるツールの紹介に留まらず、AI と共に歩むこれからのハードウェアエンジニアにとっての「思考と作業」の本質を問い直す内容となっています。最新の技術トレンドを掴み、圧倒的な開発スピードを手に入れたいすべての FPGA エンジニア必読の書です。

(インプレス NextPublishing は、株式会社インプレス R&D が開発したデジタルファースト型の出版モデルを承継し、幅広い出版企画を電子書籍＋オンデマンドによりスピーディで持続可能な形で実現しています。)

<<目次>>

第 1 章 AI 時代の FPGA 開発

第 2 章 エージェント・コーディング環境の構築

第 3 章 スキル機能

第 4 章 チームエージェント

第 5 章 FPGA の HDL コーディングテクニック - 生成 AI の失敗例

第 6 章 SystemVerilog の功罪

第 7 章 AI 時代の FPGA 開発環境

第 8 章 AI 時代のプログラミング

第 9 章 思考と作業、設計と開発

第 10 章 AI エージェントとの向き合い方

付録 A Appendix-A: Layers によるチームエージェント事例 - FPGA 向け JPEG エンコーダー開発

付録 B Appendix-B: Layers によるチームエージェント事例 - ハードウェア RTOS 内蔵 RISC-V プロセッサ開発

<<著者紹介>>

大手メーカーで LSI 設計、放送・通信機器、ハードウェア OS のベンチャー参画のあと、設計コンサルタントとして活動中。FPGA マガジン(CQ 出版)にて、執筆活動中。「ひでみのアイデア帳」(<http://sweetcafe.jp/>)にて、FPGA 関係の情報を発信中。

<<内容紹介>>

第8章 AI時代のプログラミング

本書の趣旨を実現するために必要となるプログラミングの変遷も、ソフトウェアとハードウェアの両面からレイヤー構造として分析する。バイナリからアセンブラ、C言語、スクリプト、そして自然言語へと変化する入力ソースの変遷を分析し、高位合成の必要の場について検討する。

AIが普及するとプログラミングがなくなるという大げな意見をよく聞くこともあるが、筆者はプログラミング自体はなくなりたくないと考えている。現在のプログラミングは、決まった規則に沿った言語を使用して、動作を記述していくものである。AI時代のプログラミングは単に手段が変わるだけであり、自然言語でプロンプトを用いて入力要求と出力要求を指示する方法で行っていくことになる。

その昔、バイナリで記述していたプログラムがアセンブラになり、C言語が生まれスクリプト言語へ進んだ。それぞれのレイヤーのプログラマーはたとえば、アセンブラを読むだろうか。スクリプトを書く人が言語のソースコードを読むだろうか。高位になるほど、中間レイヤーの言語は見られなくなる。AI時代では発生されたコードが今までのプログラミングの中間レイヤーにあたり、読まれることは少なくなるだろう。

ただ、現在のプログラミングでも性能をチューニングするときはアセンブラで記述したり、スクリプト用に最適化ライブラリーを作成するためにC言語やRustで記述することがある。このような機動的なプログラミングは、今後も残っていくだろう。

重要なのは、プログラミングの本質は「意図を伝えること」であるという点である。バイナリの時代は非と1で意図を伝え、アセンブラでユニークなコードで、C言語では関数と変数で意図を伝えてきた。AI時代は自然言語で意図を伝える。伝える対象がCPUからAIに変わっただけで、プログラミングの本質は何も変わっていない。

8.1 プログラミング言語のレイヤー構造

現在のプログラミングは、CPUなどで動作するまでのレイヤーとして分類できる（表8.1）。

表8.1 プログラミング言語のレイヤー構造

レイヤー	代表例	実行速度	移植性	特徴
スクリプト	Python、Ruby	低	高	実行時に1行ずつ解釈。開発が速い。
ソフトウェア言語	C、Rust	中	中	事前に1行単位でコンパイルが可能なため速い。
中間表現 (IR)	LLVM IR	高	高	最適化のための共通言語。
アセンブラ	x86_64、ARM	高	低	CPU命令と1対1で対応。
バイナリ	ELF、...	最高	無	実行の瞬間でそのままに動作形態。

8.2 ソフトウェアとハードウェアのレイヤー比較

ここはFPGAやASICの開発者のレイヤーを当てはめると、次のように表現できる（表8.2）。ソースコード設計者の観点からすると、中間表現と低級表現が中間言語的な位置づけになる（図8.1）。FPGA限定で高位合成を扱う場合もあるだろう。その場合は次のように表現できる（表8.3）。AI時代はどのような表現になるだろう（図8.2）。AI時代は低級コード、中間表現、低級表現が中間言語的な位置づけになる（表8.4）。つまり、FPGA開発でSystemVerilogなどのHDL言語、高位合成でのC/C++/Rustなどの言語は中間言語になる。もはや、高位合成の必要性がないと言わざるを得ない。さらに、中間言語に機能度上げる言語を使用する必要性がなくなってくる。

図8.1 ソフトウェアとハードウェアのレイヤー分類図

レイヤー	ソフトウェア (CPU)	ハードウェア (FPGA/ASIC)	AI時代
入力ソース / 意図設計	Python / Rust / C	SystemVerilog / VHDL	意図言語
中間コード	—	—	Python / C++ / Rust
中間表現	LLVM IR	ネットリスト	LLVM IR / ネットリスト
低級表現	バイナリ	ビットストリーム / チップ	バイナリ / ビットストリーム

表8.2 ソフトウェアとハードウェアのレイヤー分類

レイヤー	ソフトウェア (CPU/Rust)	ハードウェア (FPGA/ASIC)
入力ソース / 意図設計	Python / Rust / C	SystemVerilog / V
中間表現	LLVM IR	ネットリスト (netlist)
低級表現	アセンブリ (asm)	低級表現 (ghdl / GHDL)
最終形態	機械語 (binary)	ビットストリーム / チップ

AI時代のプログラミング変遷とレイヤー構造

第8章では、AI時代におけるプログラミングの変遷をソフトウェアとハードウェアの両面からレイヤー構造として分析します。バイナリからアセンブラ、C 言語、スクリプト、そして自然言語へと進化する入力ソースの変化を示します。高位合成での C/C++/Rust などの言語も同様に中間言語になると整理しています。

第10章 AIエージェントとの向き合い方

本書の趣旨を実現するために必要となるプログラミングの変遷も、ソフトウェアとハードウェアの両面からレイヤー構造として分析する。バイナリからアセンブラ、C言語、スクリプト、そして自然言語へと変化する入力ソースの変遷を分析し、高位合成の必要の場について検討する。

10.1 FPGA開発でのAI活用Tips

10.1.1 まずはシミュレーションを完遂することを目指す

HDLを完成させる場合、まずは小さなモジュール単位からシミュレーションの完遂を目指すように成功しやすい。

10.1.2 バイブライニングやモジュールの並列度を上げる場合

回路の性能アップのためにモジュール内での複雑なパイプラインやモジュールの並列性を上げる場合は、最初から処理性能を上げる指示をする前に一度、ワークベンチに動作するものを完成させてからパイプラインや並列度を構成する設計変更を行うのが成功しやすい。

10.1.3 VivadoやQuartusなどのIPを教えるを合戦やシミュレーションから実行

VivadoやQuartusの実行IPを指示すると、合成やシミュレーションをエージェントから実行させることが可能である。

10.1.4 一度に複数のことをやらせない

一度に複数の指示を行うと作業内容が複雑化してしまい、成功しないケースがある。

10.1.5 AIは一度、思い込んだらなかなか変えない

エージェントは一度思い込んだことをなかなか変えることがなく、二度三度と失敗した場合はずっと成功しないケースが発生する。その場合は諦めることも大切である。

10.1.6 両端できる仕組みを持つておく

二度三度と間違えるケースでも、一度リニアすることで成功するケースもある。作業のキリの良いところで作業記録を保存しておくことが大切である。

10.1.7 スキルとチームエージェントを混ぜない

チームエージェントにスキルを適用すると、偏った得意い力をするケースが多い。スキルを使用する場合は、個々のエージェントに好して適用する。

10.1.8 チームエージェントでは報道相は大切

チームエージェントを構成する場合は、個々のエージェントには作業のキリの良いときに作業記録を出力させて、エージェント間で連携させる、エージェント同士が連携しやすい。

10.1.9 気に入らなかつたら作り直せば良い

生成したコードがなんとなくいい感じに見えるが、気に入らないこともあるだろう。その場合はコードを再生成すればいい。再生成に重要なのは、それまでに作業した記録である。

10.2 AIとの効果的な対話手法

AIで質問に対して返答したものが能力されないということをよく耳にし、AIを使うことに懐疑的な意見を持つことも多い。その原因はすごく簡単で、単に指示が足りていないだけである。AIは質問から出力を生成するのではなく、質問から出力を導き出していることを覚えておかない。それでは、どうやってAIの能力を刺激したものへ刺激を上げていくかというと、意図した方向に導くように質問の具体性を上げるしかない。

筆者の経験上、現在のモデルは同じ質問に対して3回繰り返して質問するよという感覚がある。AIはコーディングで悩んでいる作業より耳にしますが、それと同じくらい質問の精度を上げるために質問も3回繰り返すようなものである。今後はこのとおりであるとは断言できない。AI機能は常に進化しているので、すでにエスパーなAIも登場するかもしれないが、現状は良い結果を導き出している。

質問を3回繰り返す手法は、「同じ質問を3回繰り返す」と「3回の質問で具体性を上げていく」の2種類の存在。

10.3 同じ質問を3回繰り返す

異なるセッションで同じ質問を3回繰り返す、出力結果を見て、一番良い出力を使用して作業を進める。複数の質問は、3回の出力が微妙に異なってくる。同一セッションで3回同じ質問をする上結果の精度が上がるといえるが、この3回質問する手法はセッションである。同一セッションでの質問、質問したコンテキストが残っている、その回答に学習した回答が返ってくる。

たとえば、「数字の合計を計算するスクリプトを作成する」という質問を例に取ると、出力として次のスクリプトを作成するケースが考えられる。
・Pythonで作成する
・TypeScriptで作成する

AI エージェント活用 Tips と 3 回質問法

第10章では、FPGA 開発での AI 活用 Tips を解説します。シミュレーション完遂や複数指示禁止など 9 項目を示します。AI は一度思い込んだことを忘れない傾向があります。作業記録の保存で再開できる仕組みを持つべきです。同じ質問を 3 回繰り返す手法が有効です。3 回の質問で具体性を上げていく方法も示します。

<<販売ストア>>

電子書籍：

Amazon Kindle ストア、楽天 kobo イーブックストア、Apple Books、紀伊國屋書店 Kinoppy、Google Play Store、honto 電子書籍ストア、Sony Reader Store、BookLive!、BOOK☆WALKER、BOOK TECH

印刷書籍：

Amazon.co.jp

※ 各ストアでの販売は準備が整いしだい開始されます。

※ 全国の一般書店からもご注文いただけます。

<<技術の泉シリーズについて>>

『技術の泉シリーズ』は、技術者の知見のアウトプットである技術同人誌を底本とした、2017 年創刊の技術書シリーズです。NextPublishing によるスピーディな編集制作とプリントオンデマンドによる 1 冊からの印刷製本により、技術の変化に追従しつつ返品や品切れのないサステナブルな出版モデルを特徴としています。本シリーズを通じて、エンジニアの“知の結晶”である技術同人誌の世界に、より多くの方が触れていただくきっかけとなることを目指しています。



【株式会社インプレス】 <https://www.impress.co.jp/>

シリーズ累計 8,000 万部突破のパソコン解説書「できる」シリーズ、デザイン・グラフィック関連の書籍ブランド「MdN」、「デジタルカメラマガジン」等の定期雑誌、IT 関連の専門メディアとして国内最大級のアクセスを誇るデジタル総合ニュースサービス「Impress Watch シリーズ」等のコンシューマ向けメディア、「IT Leaders」をはじめとする企業向け IT 関連メディアなどを総合的に展開・運営する事業会社です。IT・デザイン関連出版メディア事業、およびデジタルメディア&サービス事業を幅広く展開しています。

【インプレスグループ】 <https://www.impressholdings.com/>

株式会社インプレスホールディングス(本社:東京都千代田区、代表取締役:塚本由紀)を持株会社とするメディアグループ。「IT・デザイン」「音楽」「山岳・自然」「航空・鉄道」「モバイルサービス」「学術・理工学」を主要テーマに専門性の高いメディア&サービスおよびソリューション事業を展開しています。さらに、コンテンツビジネスのプラットフォーム開発・運営も手がけています。

【NextPublishing】

NextPublishing は、インプレス R&D が開発した電子出版プラットフォーム(またはメソッド)の名称です。電子書籍と印刷書籍の同時制作、プリント・オンデマンド(POD)による品切れ解消などの伝統的出版の課題を解決しています。これにより、伝統的出版では経済的に困難な多品種少部数の出版を可能にし、優秀な個人や組織が持つ多様な知の流通を目指しています。

【本件に関するお問合せ先】

株式会社インプレス : PubX 室

E-mail: np-info@impress.co.jp