HITACHI
Inspire the Next

M言語とSimulinkモデルについて

C言語やHDLなどの機械向けの言語を使用した開発環境から、M言語やSimulinkのブロック線図を使った、より開発者が理解しやすい開発環境に移行することで、作業・レビュー・教育の効率が向上します。

M言語について

M言語(MATLAB言語)は、数値計算プログラムの記述に広く用いられている言語の一つで、行列を一つの変数で表すことができるため、行列計算を簡潔に記述できるのが大きな特徴で、対話的に変数やパラメータを即時に変更して結果を確認することもできます。その他、一般的な特徴についてC言語と比較して以下の表に示します。

No.	比較項目	C言語	M言語
1	コード量	多い(※)	少ない
2	行列演算	制御文を含めて記述(※)	1行で記述可能
3	固定小数点	表現不可(※) ・開発者が仮想的に決めて整数型にて演算 ・丸め飽和を演算式で表現 ・型の伝搬はなし	表現可能 ・固定小数点型を使って記述可能 ・丸め飽和を変数に対して指定可能 ・演算ルールに従って型が自動伝搬
4	数学関数	独自に用意するかライブラリを入手・選択する必要がある	多くのものがあらかじめ用意されている
5	実行環境	無償入手可能 ・ただしコンパイラやライブラリのバージョン等を 合わせておく必要がある	MATLABが必要 ・MATLAB Compilerによりスタンドアロン化可能 ・MATLAB CoderによりC言語に変換可能 ・MATLABのバージョンは合わせておく必要がある
6	ステップ実行	統合環境により可能 ・変数の値は参照および変更可能	標準で利用可能 ・ブレーク中にコマンドラインで解析可能
7	データの可視化	個別に用意する必要がある	豊富な機能の中から選択して使用できる

※ ライブラリを使用することにより改善することは可能。

Simulinkについて

Simulinkはモデルベースデザインのためのブロック線図環境であり、ハードウェアに実装する前に、システムをデザインしてシミュレーションすることが可能です。Simulinkモデルを使用したハードウェアの論理設計の特徴について、従来手法と比較して以下の表に示します。

No.	比較項目	従来手法(回路図＋HDL)	Simulinkモデル(ブロック線図)
1	操作性	特化していないため悪い ・階層的に操作不可(ページまたぎ)	特化しているため良い ・階層的に操作可能
2	可読性	悪い ・HDLは処理の流れ通りに記述困難	良い ・処理の流れを素直に表現できる
3	診断機能	なし	あり(人手の誤りを静的診断で検出できる)
4	共通化	まとまった単位でのみ可能 ・モジュール	様々なスケールで可能 ・モジュール > モデル > ブロック
5	固定小数点	表現できない、自動桁合わせ・伝搬しない ・設計者がすべての変数および演算に注意を払う必要がある	表現できる、自動桁合わせ・伝搬する ・設計者はポイントを押さえるだけで良い
6	属人性	出やすい	出にくい
7	レビュ	非効率 ・回路図とコードで2回のレビュが必要 ・指摘の修正がレビュ後となるため確認が別途必要	効率的 ・モデルレビュの1回のみ ・指摘をその場で修正・確認できる
8	保守性	回路図とHDLの両方	Simulinkモデルのみ
9	設計資産	保守・流用しにくい	保守・流用しやすい
10	教育	立ち上がりが遅い ・回路イメージとHDLの対応の学習が必要 ・HDLを回路図との対応を図りながら説明する必要がある。 ・シミュレーション実行までのステップが長い	立ち上がりが早い ・HDLの学習が不要 ・回路イメージでダイレクトに議論が可能 ・シミュレーションが容易に実行できるため理解が早い

製品に関する詳細・お問い合わせは、営業担当員または下記へ

株式会社 日立情報通信エンジニアリング

神奈川県横浜市西区みなとみらい2－3－3
クイーンズタワーB　25階　　〒220－6122
営業統括本部 第2本部 エンジニアリング戦略部

お問い合わせは下記ページから
お問い合わせフォームにお進みください。

モデルベース開発(MBD)サービス
https://www.hitachi-ite.co.jp/products/mbd/index.html

© Hitachi Information & Telecommunication Engineering, Ltd. 2024. All rights reserved.