



# システム LSI 開発を成功させるには

開発者の連携を強める三つのポイント

穴田 啓樹

## 1. はじめに

2002年7月に開催された第8回 ZIPC ユーザーズカンファレンスで、渡辺政彦副社長が、今後の展望を示すべく、ZIPC ロードマップを説明されました。そのロードマップには、「システム LSI 対応」「プロジェクトマネジメント構想」が掲げられています。

システム LSI 対応については、システム LSI 向け仕様合成ツール「XModelink」という答えが示されています。プロジェクトマネジメント構想については、詳細な説明がありませんでしたので、現時点ではわかりません。第9回 ZIPC ユーザーズカンファレンスで発表されるものと思います。

さて、システム LSI 開発は、開発規模が大きいので、プロジェクトマネジメントが不可欠です。そこで、本稿ではプロジェクトマネジメントの視点から、システム LSI 開発を成功させるポイントについて私見を述べ、参考に付したいと思います。

## 2. 開発を成功させるポイント

システム LSI 開発では、開発者間の依存事項が多いので、開発者の連携を強める必要があります。ここでは、重要と思える三つのポイント

- ・ 組織構造の整備
- ・ 思惑マネジメント
- ・ 開発工程の相互理解

について説明します。

### 2.1 組織構造の整備

システム LSI 開発では、開発者の連携を強めるために、従来の組織にとらわれない、開発効率のよい組織構造を整備することが重要です。米国プロジェクトマネジメント協会 (PMI) が発行しているプロジェクトマネジメントの知識体系 (PMBOK) [1]では、組織構造を三つの基本形、「機能型、プロジェクト型、マトリクス型」に分類しています。どの型の組織がシステム LSI 開発に適しているかは、事業環境に依存するので簡単には説明できませんが、私はプロジェクト型組織の開発効率がよいと思います。

このプロジェクト型組織を選ぶ理由は三つあります。

- ・ 部門間の利害にとらわれない
- ・ 一つの場所に集結して作業する
- ・ プロジェクトマネージャーが指揮しやすい

第一に、部門間の不要な交渉が減るので、効率がよい点です。たとえ同じ企業といえども、それぞれの部門には独自の文化があるものです。システム LSI のように関連部門が多い場合には、文化の不協和が障害になります。

第二に、開発者が不明確な点を確認しやすいので、効率がよい点です。例えば、ちょっとした仕様の解釈を確認したいときがあります。その場で確認できるか否かは、積み重なると仕事の効率に大きな影響を与えます（コラム1）。

第三に、プロジェクトマネージャーが組織全体の采配をふることができるので、効率がよい点です。例えば、機能型組織

では、プロジェクトマネージャーは部門リーダーを通じた間接采配になりがちですが、それでは問題の解決が遅れます。

以上の三点から、私はプロジェクト型組織がよいと考えます。

## 2.2 思惑マネジメント

システム LSI 開発では、開発者の連携を強めるために、プロジェクトマネージャーが、LSI 設計者とソフトウェア開発者それぞれの思惑を聞き出し、利害を調整することが重要です。ここでの思惑とは、開発者が常識として持ち合わせている、言葉や文書で表現されない価値観や利害、要求のことです。なお、思惑マネジメントは私の造語であり、PMBOKには含まれていません。思惑はプロジェクトのリスクになり得ます。このリスクを防ぐために、前節のプロジェクトマネージャーが、事前に思惑を聞き出し、関係者の利害を調整する必要があります。

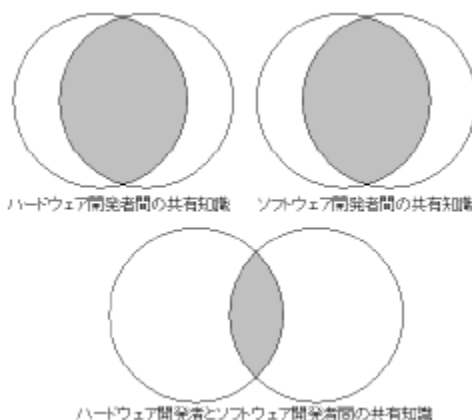
### コラム1 “会話”のすすめ

会話より電子メールの方が手軽なので、電子メールで連絡をとることが多い方もいると思いますが、大事な話題は会話で伝えることをおすすめします。

たしかに、電子メールは読み手の都合に合わせて読めるという点が便利です。しかし、電子メールの情報伝達効率は、会話に劣ります。内容を伝えるには、伝えたい内容に加え、発信者のニュアンスが必要ですが、電子メールではこのニュアンスが伝わりにくいのです。

例えば、電子メールでの不毛な水掛け論の後に、会話したら誤解が解けた、ということがあります。これを裏付けるデータとして、心理学者 Albert Mehrabian 博士による「Total Liking」[2] があります。好意が伝わる時の情報は非言語系が 93%を占めている、という研究です。好意が伝わればコミュニケーションしやすいのは明らかですが、全て言語系で構成される電子メールで好意を伝えるのはむずかしいものです。したがって、大事な話題は、電子メールで伝えるのではなく、会話で伝えるべきです。

例えば、ハードウェアとソフトウェアのグループに分かれて製品を開発するとします。グループ内では、開発者間の共有知識が多いので、簡単な説明でも伝わります。一方、グループ外とは、開発者間の共有知識が少ないので、グループ内と同じ説明では伝わりません（**図 1**）。（**コラム 2**）聞き手が、不足する情報を聞き手の期待、すなわち、思惑で解釈すると、誤解の原因にもなります。そこで、プロジェクトマネージャーは、隠れた思惑を探し出し、問題が発生する前に関係者の利害を調整する必要があります。

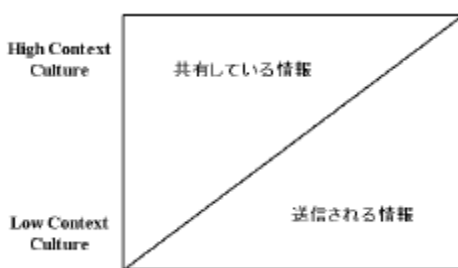


**図 1** 開発者間の共有知識

### コラム 2 コミュニケーションモデル

コミュニケーションモデルを理解していると、相手の情報の受け取り方がわかるので、スムーズに情報伝達できます。日本のコミュニケーションモデルでは「一を聞いて十を知る」といことが言われます。残りの九の情報話し手と聞き手が共有しているので、聞き手が補完できるのです。うまく伝わる場合のコミュニケーション効率は高いのですが、話し手がこれを期待してしまうと伝わらない場合もあります。皆さんは、「それは常識だろう」と言ったことはありませんか？それは、共有していない知識を前提にコミュニケーションした結果、発生したギャップです。

このようなコミュニケーションモデルは、人類学者 Edward T. Hall 博士によって研究されています。著書 "BEYOND CULTURE" [3]では、共有知識が多く送信される情報の少ない文化を High Context Culture、共有知識が少なく送信される情報の多い文化を Low Context Culture と分類し、両者の送信される情報量の差から、コミュニケーションギャップが発生すると分析しています（**図 2**）。日本は High Context Culture ですが、Robert Hilke 氏は、日本の中でも小さな文化の違いがあるので、共有していない知識を前提にコミュニケーションしてギャップが発生しないように注意する必要があります。[4]



**図 2** 文化による情報の扱いの違い

ハードウェア開発者とソフトウェア開発者の間の理解不足が原因で問題が発生することは、皆さんも経験されているでしょう(コラム3)。これが、LSI 設計者とソフトウェア開発者の間では、ますます理解が不足するようです(ここでは、基板を開発する担当者をハードウェア開発者と表現し、LSI 設計者と区別しました。)

両者の理解を促そうとする動きもあります。名古屋大学の高田広章教授によれば、両者の交流を図るために、毎年7月に「DA シンポジウム」(DAS)と「組込みシステム技術に関するサマーワークショップ」(SWEST)を同日開催しているそうです。DAS と SWEST は、相互にチュートリアルや分科会に参加できるシステムになっています。この機会を活かすべく、2002 年の DAS 「C 言語設計部屋」に参加してみたところ、両者にはギャップがあり、カルチャーショックを受けました。このカルチャーショックが、高田教授の意図された交流の第一歩かもしれません。

効率よくシステム LSI を開発するには、技術の協調だけではなく、人の協調も必要です。この協調の必要性について、イーソル株式会社の上山伸幸氏とお話しさせて頂いたところ、同意して下さり、両者のギャップを埋めるべく尽力されている旨、説明していただきました。上山氏は、1998 年に次のように書かれています。

「**Embedded System 設計文化のギャップ**  
Co-Design を行うためには、ハードウェア(半導体)、ソフトウェア両方の知識が要求されます。しかしながら、この二つの業界にはかなりのギャップがあります。」[5](一部を抜粋)

このような根の深い問題を無視すれば、仕事に支障をきたすのは明らかです。そこで、プロジェクトマネージャーが、開発者間のギャップになるような思惑を聞き出し、問題が発生しないように利害を調整することが重要なのです。

### コラム3 開発者の気質

「ソフトウェア開発者は論より RUN だ。」そんな興味深い話を、2003 年 1 月に開催された第 5 回 CEST 技術セミナー「C 言語ベース設計と高位合成技術」で聞きましたので紹介します。どなたの発言か失念して申し訳ありませんが、その方の話によると、C 言語設計ツールをハードウェア開発者に与えると、一度で完成度の高い回路を合成するそうです。それに対してソフトウェア開発者は、シミュレーターで動作確認しながら何度もチューニングを繰り返して完成度を高めていくそうです。良し悪しは別として、少なからず、そのような傾向があるのではないのでしょうか。

## 2.3 開発工程の相互理解

システム LSI 開発では、開発者の連携を強めるために、LSI 設計者とソフトウェア開発者が、開発工程をお互いに理解することが重要です。特に、ソフトウェア開発者が LSI 開発工程を理解する必要があります。なぜならば、LSI 製造は厳密に工程管理されているので、ソフトウェア開発工程が LSI 製造工程に合わせる必要があるからです。なかでも、ソフトウェア開発工程から見たハードウェア仕様の確定時期が、従来に比較して早まる点には注意する必要があります。

それでは、ハードウェア仕様の確定時期が早まるのが、ソフトウェア開発に与える影響を考えてみましょう。LSI は大量生産するので、コストを抑えるために、システムの最適性能で設計することが望まれます。しかし、LSI 設計時点ではソフトウェアが完成しておらず、必要性能が不明確なまま LSI 仕様を確定しなければならぬケースもあります。そして、ソフトウェアが完成した時点で性能不足がわかり、慌てるわけです。もし、LSI 仕様を変更すれば、時期と内容によって、多額の費用と時間がかかってしまいます。

この事態を防ぐには、ソフトウェア開発者が LSI 設計上のマイルストーンを理解して、必要な時期に必要な設計情報を LSI 設計者に提供する必要があります。

ここでの設計情報とは、例えばメモリに関して、どんなバスに、どんな性能で、どんなサイズの、どんなメモリを接続するかといった、LSI 仕様です。これからの組込みソフトウェア開発者は、ハードウェアを与えられるのを待つのではなく、LSI 設計工程を理解して、性能設計に積極的に加わるべきです。

## 3. おわりに

システム LSIこそ日本企業の生き残る道、といった特集番組が放映され、世間のシステム LSI への期待はかなり高いようです。システム LSI 開発を成功させるには、開発者間の連携が重要です。これからシステム LSI を開発される方は、これまでに説明した三つのポイント、組織構造の整備、思惑マネジメント、開発工程の相互理解について取り組み、連携を強化することをお奨めします。きっと、スムーズに開発できるでしょう。

最後に、ZIPC WATCHERS に寄稿させて頂くのは今回で三度目になりますが、このような機会を与えて頂いたキャッツ株式会社様に感謝いたします。次回は再び ZIPC 活用事例を紹介したいと思えます。

あなだ・けいじゅ ヤマハ株式会社

## 参考文献

- [1] Project Management Institute, "A Guide to the Project Management Body of Knowledge (PMBOK Guide) 2000 Edition", Project Management Institute, 2000
- [2] Albert Mehrabian, "Silent Messages", Wadsworth Publishing, 1981
- [3] Edward T. Hall, "BEYOND CULTURE", ANCHOR BOOKS, 1976
- [4] Robert Hilke, "日本人のための国際人養成講座", 株式会社アルク, 2001
- [5] 上山伸幸, "Embedded System 設計手法の基礎知識", CQ 出版社, Interface 1998 年 5 月号