このレポート課題において、課題4を選択肢した。

# 記述したソースコード、実行結果

記述したソースコードは、その変更部分をレポート末尾に別途添付した。

以下、簡単に変更点を説明する。

・鍵生成部は2段、暗号復号部は4段とし、暗号復号部はパイプライン化を行った。

・13, 14行目:

それぞれ鍵生成部と暗号復号部を多段にするための状態変数を宣言している。

・key\_state: 鍵生成部の2bitの状態変数

・data\_state: 暗号複合部の4bitの状態変数

・19～23行目:

keyin, datin信号が来たとき、それぞれkey\_state, data\_stateを初期化している。

・25～77行目:

鍵生成部の処理。key\_stateの値によってどの段の処理を行うかを分けている。

・79～121行目:

暗号復号部の処理。dat\_stateのあるビットがセットされていれば、それに対応する段の処理を行うというようにして、パイプライン化を行っている。

テストベンチの実行結果は次ページの図1に示す。

テストベンチにおいては、p1, p2, p3とランダムな数値を発生させ、それを暗号化した結果e1, e2, e3を得ている。

p1, p2, p3をクロックが変わるたびに暗号部に入力しても、それぞれその4クロック後にe1, e2, e3が得られているので、4段のパイプラインとして動いていることがわかる。

また、e1, e2, e3を復号化（同じキーで暗号化）した結果ee1, ee2, ee3を得ると、それぞれp1, p2, p3と一致していることがわかるので、des暗号として正しく動作していることがわかる。

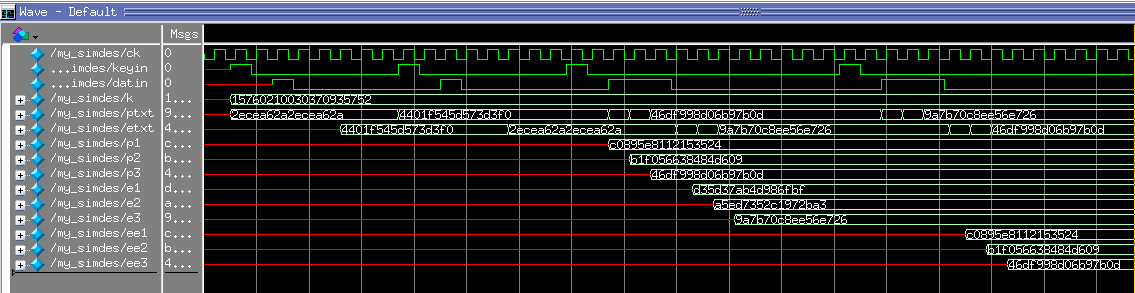


図1. テストベンチの結果

# 考察

## 2.1 実行時間

(I) 鍵生成部

転地1回あたりの実行時間を, シフト演算1回あたりの実行時間をとすると、

(II) 暗号復号部

転地1回あたりの実行時間を, XOR演算1回あたりの実子時間をとすると、

## 2.2 ハードウェア量

(I) 鍵生成部

ビットの転地に必要なハードウェア量を、ビットのシフト演算に必要なハードウェア量をとすると、

(II) 暗号復号部

ビットの転地に必要なハードウェア量を、ビットのXOR演算に必要なハードウェア量をとすると、

## 2.3 論理合成結果

段数を増やしたときのクロックサイクルと面積の変化を考える。

クロックサイクルは、段数がn段になると、1段のときの1/nになる。

面積は、段数が増えてもほとんど変わらないか、値を保持するフリップフロップの分だけ微増する。

暗号復号部の段数を増やし、実際に論理合成した結果を表1に示す。

段数を増やしたとき、クロックサイクルは段数におおよそ反比例するように減少し、面積はほとんど変化していないことがわかる。

表 1暗号復号部の段数を増やしたときのクロックサイクルと面積

|  |  |  |
| --- | --- | --- |
| 暗号復号部の段数 | クロックサイクル | 面積 |
| 1 | 100.14 | 17.64 |
| 3 | 46.06 | 16.6 |
| 4 | 33.76 | 16.43 |
| 6 | 31.12 | 17.25 |
| 8 | 18.42 | 17.15 |
| 16 | 11.16 | 19.15 |