マルウェア検出処理のハードウェア化の検討

寶満 剛　渡邊 誠也　名古屋 彰
岡山大学大学院　自然科学研究科

1. 緒言

近年，コンピュータウィルスやワーム等，マルウェアと呼ばれる悪質なソフトウェアの流通が問題となっており，ソフトウェアによってマルウェアを検出することにより被害を防止する手段が一般的に用いられている．しかしマルウェアの増加に伴い処理時間は増大し，他の実行プログラムの動作にも影響を及ぼすことが問題となる．そこで本研究では検出に必要なパターンマッチング処理を再構成可能ハードウェアで実現することにより，処理を効率化する手法を検討する．

2. パターンマッチング処理のハードウェア化

一般にマルウェア検出は，シグネチャと呼ばれるマルウェアのパターン情報を記録したファイルと，検査対象のデータとのパターンマッチング処理で行われる．本研究ではアンチウィルスソフトウェアの一つである ClamAV(Clam Antivirus)[1]のシグネチャを用いる．

パターンマッチング処理のハードウェア化の方法には，ルータ等における侵入検知システムのパターンマッチング回路生成に用いられる OHE（One-Hot Encoding）方式[2]が提案されている．回路の基本構成は図 1 のようになる．大きく分けてパターン情報と入力データを比較する比較部と，パターンマッチング履歴を保持する FF(FlipFlop)部の二つからなる．FF 部は同時刻で複数のマッチング途中段階の履歴を持つことが可能である．FF 部の状態は前状態と比較部の結果によって遷移し，最終段の FF でマッチング結果が出力される．以下，FF と AND ゲートのセットをパターンマッチングの「単位回路」と呼ぶことにする.また，回路規模の増加を避ける為に，先頭に同一パターンを有する複数のシグネチャに対応する単位回路は共有して構成されるようになっている．

3. 正規表現に対応した効率的単位回路の提案

ClamAV のシグネチャには正規表現で表されるものも含まれる為，前節で示した基本構成だけでこの表現形式に直接対応することは出来ない．OHE 方式を提案した Sidhu ら[2]により，選

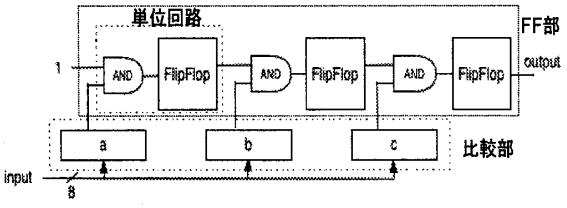

図 1 “abc”を検出するパターンマッチング回路

択を表す|と 0 回以上の繰り返しを表す*の二つの正規表現を実現する回路が示されているが，ClamAV にはその他に回数制限のある繰り返しを表す{-*n*}, {*n*}, {*n*-*m*}, {*n*-}が存在する．これらはそれぞれ *n* 回まで，*n* 回，*n* から *m* 回，*n* 回以上の繰り返しを表す．例えば a{-3}は{null,a,aa,aaa}を表す．これを単純に回路にすると，単位回路を各要素の数だけ並べたものになるが，中には *n* が数千以上になる場合も存在し，ハードウェアのリソースが増大してしまう．そこで本研究では例えば図 2 のようなカウンタを含む単位回路を提案する．この単位回路は繰り返しの回数に応じて次段への信号を変化させることが出来る．このような単位回路を用いて上述の 4 種の正規表現を実現する．

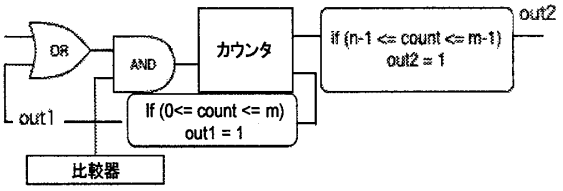

図 2 カウンタを含む単位回路

4. 先頭以外の同一パターンに対応した回路共有

ClamAV のシグネチャパターンの先頭を調べると，シグネチャ間での同一パターンが少ないことがわかる．そこで従来の手法とは別に，先頭以外の同一パターンに対応する単位回路を共有させる手法について検討する．共有は図 3 のように行う．先頭以外の単位回路を共有する為，どの単位回路から信号が遷移したかを記憶する必要がある．また共有部の終端へ信号が遷移するまで保持している値が変化してはならない為，共有したいパターンの接頭辞が他のパターンの

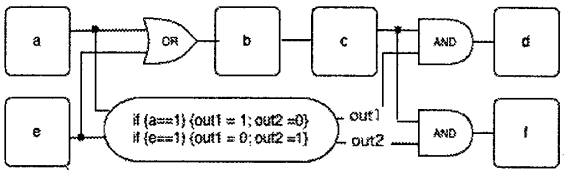

図 3 “abcd”と“ebcf”の単位回路共有

共有部分と接頭辞を合わせた中に含まれていると共有出来ない．図 3 では a は ebc に，e は abc に含まれていない為，共有が可能となる．

共有パターンの探索は上述の条件を満たしたものの中で，(共有部の単位回路数)×(共有可能なシグネチャの数-1)の最も多いものを選ぶ．

5. 実装結果と評価

シグネチャからパターンマッチング回路の VHDL 記述を生成するプログラムを作成し，Xilinx の ISE9.2i 評価版を用いて FPGA への論理合成を実施した.ターゲットデバイスは Virtex-Pro XC2VP100 である．

実際のシグネチャに対応する回路を構成する前に，先ず，カウンタを含む単位回路を用いた場合とそうでない場合で，正規表現 ab{-*n*}c を回路化した際の Slice の使用数の比較を行った結果を表 1 に示す．ClamAV では *n* が数百以上のシグネチャが多数存在することから，カウンタ無しの実現は困難と判断し，以下ではカウンタ有りの実現で議論する．

表 2 はソフトウェアである ClamAV と先頭の同一パターンのみの共有を行った回路で，それぞれ任意の入力データに対するパターンマッチング処理をした場合の処理性能の比較である.双方ともランダムに 100～500 パターンずつ抽出した同じシグネチャを使用した．ClamAV を動作させた環境は CPU：PentiumⅢ 800MHz，メモリ：512MB である．生成されたパターンマッチング回路は 1 クロックサイクル当たり 1byte のデータを処理する.ここでは 1GB のデータを入力データとした場合の処理時間から，1 秒間当たりの処理データ量を算出し比較した．パターンマッチング回路は ClamAV の約 20 倍の処理能力があることがわかる．

表 1 ab{-*n*}c に対応する回路の Slice 数の比較

n	カウンタ無し	カウンタ有り
10	12	9
100	68	16
1000	633	18
10000	6295	23

表 2 ClamAV との処理性能比較

シグネチャ数	ClamAV(MB/s)	生成回路(MB/s)
100	10.006	219.4
200	9.898	224.6
300	9.491	177.9
400	9.629	177.9
500	9.593	175.8

表 3 実装面積結果(Slice 数とその使用率)

シグネチャ数	共有前	共有後
100	4914(11%)	4858(11%)
200	9558(21%)	9238(20%)
300	13899(31%)	13421(30%)
400	19191(43%)	18400(41%)
500	26346(59%)	22477(50%)

表 3 は先頭パターンの共有前後のターゲットデバイスにおける Slice の使用数と Slice の総数 44096 に占める割合をシグネチャ数毎に示している．シグネチャ数が増えるほど共有による削減率は大きくなっている．しかし 500 パターンで Slice の使用率が 50%に達していることから，実用的なシグネチャ数（数万パターン以上）に対応するためにはさらに実装面積を削減する手法が必要である．

なお表には示していないが，シグネチャの先頭以外の同一パターンも共有した場合，さらに約 2%の単位回路が削減できることを確認している．

6. 結言

ソフトウェアによるマルウェア検出処理の問題から，処理のハードウェア化の手法について検討し，ClamAV のシグネチャを用いて回路記述を生成，論理合成結果から性能や実装面積を評価した．

ClamAV よりも数十倍の処理能力の向上が可能であることを確認できたが，FPGA におけるリソースの使用率が大きく，全てのシグネチャに対応するパターンマッチング回路を 1 チップ上へ実装することは困難であることが明確になった．その為，さらに面積効率を上げる手法の開発が今後の課題である．

参考文献

[1]Clam Antivirus http://www.clamav.net/

[2]Sidhu, R. and Prasanna, V.K.: Fast Regular Expression Matching using FPGAs, Proc. IEEE FCCM 2001, pp.227-238, April 2001