A New Approach to the Problem of PLA Partitioning Using the Theory of the Principal Lattice of Partitions of a Submodular Function

Subir Roy*  H Narayanan†

Abstract

An area efficient 2 level implementation of combinational logic can be achieved by partitioning the original PLA into several PLAs's each of which interacts with the others weakly. A PLA implementing a sum of products logic functions can be modelled through a bipartite graph $BG$, which specifies the intersection of rows (minterms) with columns of the AND plane (primary inputs) and the OR plane (primary outputs) respectively. We show how to achieve a good PLA partition by using the Principal Lattice of Partitions of the incidence function of $BG$.

1 Introduction

The approach of partitioning as a methodology for handling complexity inherent in the design of VLSI chips is well cited in the literature. Partitioning has been applied to various areas such as placement, logic partitioning, PLA partitioning, finite state machine decomposition etc. Partitioning can be obtained by modelling the problem in hand as a graph and iteratively finding clusters in this graph based on heuristic algorithms [1], [2]. In this paper we discuss a new approach to partitioning which detects optimal partitions which are in a (precisely stated) sense natural to the given graph.

PLA's are important in the automated synthesis of VLSI digital circuits. Its regular structure which enables automated synthesis also entails considerable wastage of silicon real estate. Moreover for a given technology there might be restrictions on the realizable size of a PLA. Topological row and column folding have been employed to minimize PLA areas. Another approach is to partition the logic functions and $E$ is the set of edges joining an input (primary inputs) and the output (primary outputs) respectively. We show how to achieve a good PLA partitioning by using the Principal Lattice of Partitions of the incidence function of $BG$.

2 PLA Partitioning & PLP

Essential to the technique of PLP is a set function which is submodular. We use $L(G): 2^V \rightarrow R$, the incidence function of $BG$ as defined below. (We use $L(\cdot)$ when it is clear from context that the underlying bipartite graph is $BG$).

Let $X$ be a subset of $V$, $L(X) =$ Number of vertices in $V_i$ which are joined by edges to some vertex in $X$. $L(\cdot)$ can be shown to be a submodular function, i.e., $L(X) + L(Y) \geq L(X \cup Y) + L(X \cap Y), \forall X, Y \subseteq V_i$, $L(\cdot) - \lambda$, for $\lambda$ real, is also submodular. Consider a partition $\Pi$ of $V_i$ into blocks $R_1, \ldots, R_k$. Define

$$(L - \lambda)(\Pi) = \sum_{R_i} L(R_i) - k\lambda.$$

Suppose $\Pi = \{R_1, \ldots, R_k\}$ minimises $(L - \lambda)$. Let $\Pi' = \{R_1', \ldots, R_k'\}$ be another partition of $V_i$ of size $k$. Then $$(L - \lambda)(\Pi) \leq (L - \lambda)(\Pi') \text{ i.e.}$$
\[ \sum_{R_i} L(R_i) - k \lambda \leq \sum_{R_i'} L(R_i') - k \lambda, \]

\[ \sum_{R_i} L(R_i) \leq \sum_{R_i'} L(R_i'). \] Thus, we say that among all partitions of size \( k \) blocks, \( \Pi \) results in the minimum value of \( \sum_{R_i} L(R_i) \). The overlap in \( V_i \) with respect to \( \Pi \) is given by

\[ \sum_{R_i} L(R_i) - | V_i |. \]

Since \( | V_i | \) is a constant, we have that

\[ \sum_{R_i} L(R_i) - | V_i | \leq \sum_{R_i'} L(R_i') - | V_i |, \] i.e. \( \Pi \) also minimises the overlap in \( V_i \).

This in turn means that among all partitions of the given PLA into \( k \) PLA's, \( \Pi \) is the one that results in the least number of overlapping amongst the primary inputs and outputs, i.e. fewer primary inputs will have to be fed to more than one AND planes of the smaller PLA's and fewer outputs will be needed to be obtained by disjunctively combining the outputs from the OR planes of different partitioned PLA's. This is shown in fig. 2b where a minimal overlap in the set of primary outputs implies a smaller sized OR plane. In a similar manner, one could individually partition the AND plane of a PLA by choosing \( V_i \) to consist of the primary inputs, partitioning it and choosing the incidence function to be \( R(\cdot) : 2^{|V_i|} \rightarrow R \). The overlaps in the subsets of \( V \) implies that there are minterms which need primary inputs from more than one block in the partition \( \Pi \) of \( V \). Since we partitioned the set of primary inputs we have to realise partial minterms from each block of \( V \) and later combine them conjunctively to obtain the actual set of minterms as shown in fig. 2c. An optimal partition of \( V \) would then result in AND being smaller. The measure of overlap is thus reflected in the size of OR and AND respectively and minimising the overlap results in their sizes being smaller.

3 Qualitative description of PLP

The PLP of \( L(\cdot) \) is a compact description of all partitions \( \Pi \) over which \( (L - \lambda) \) reaches a minimum for any real \( \lambda \). We give below a qualitative description of the PLP in order to bring out its relevance to the problem in hand.

Let \( \Pi_1, \Pi_2 \) be partitions of \( V \). We say \( \Pi_1 \geq \Pi_2 \) if every block of \( \Pi_2 \) is contained in some block of \( \Pi_1 \). \( \Pi_0 \) denotes the the partition of \( V \) into singletons while \( \Pi_1 \) the partition with \( V \) as its only block.

1. For any given \( \lambda \), there is a unique maximal partition \( \Pi_{\lambda_{\text{max}}} \) & a unique minimal partition \( \Pi_{\lambda_{\text{min}}} \) at which \( (L - \lambda) \) reaches a minimum.

2. The partition \( \Pi_3 \) at which \( (L - \lambda) \) minimum form a lattice.

3. If \( \lambda_1 \geq \lambda_2 \) then \( \Pi_{\lambda_2} \leq \Pi_{\lambda_1} \).

4. Since we are dealing with finite sets there are only a finite number of values of \( \lambda \) (called the Critical Values) at which \( \Pi_{\lambda_{\text{max}}} \) changes when \( \lambda \) ranges from \( -\infty \) to \( +\infty \). The sequence \( \Pi_0, \Pi_1, \ldots, \Pi_n \) such that \( \Pi_0, \Pi_1, \ldots, \Pi_{n-1} \) are the minimum partitions for the decreasing sequence of critical values is called the Principal Sequence of \( L(\cdot) \).

A good set of partitions to examine for PLA partition would simply be the \( \Pi^n_{\lambda_{\text{max}}} \) for the critical values. This number would usually be very much less than \( | V | \). The PLP of \( L(\cdot) \) can be generated using a network flow based algorithm of complexity \( O(m^3) \). A natural approach to the problem of PLA partition using PLP would be to, (1) Get the \( \Pi^n_{\lambda_{\text{max}}} \)'s for all the critical values, (2) Select the one, say \( \Pi^n_{\lambda_{\text{opt}}} \) whose number of blocks is closest to the desired value, and (3) If necessary redistribute vertices in different blocks of \( \Pi^n_{\lambda_{\text{opt}}} \) (if their sizes differ substantially) without increasing \( \sum L(R_i) \) excessively.

4 An Example

For the bipartite graph shown in fig. 3 and its incidence function \( R(\cdot) : 2^{|V_i|} \rightarrow R \), the Principal Sequence of partitions can be verified to be the following

\[ \Pi_0 = \{(a,b),(c,d),(e,f,g)\}, \]
\[ \Pi_1 = \{(a,b),(c,d),(e,f,g)\}, \]
\[ \Pi_2 = \{(a,b),(c,d),(e,f,g)\}, \]
\[ \Pi_3 = \{(a,b),(c,d),(e,f,g)\}. \]

The corresponding decreasing sequence of Critical Values are 2, 1.5 and 1.333 respectively.

5 Outline of Algorithms

We briefly sketch the outlines of the algorithms for the PLP of the function \( L(\cdot) \). Details may be found in [5].

We first introduce some necessary notation.

Let \( E = \{e_1,\ldots,e_n\} \) be a set and \( A = \{e_1,\ldots,e_k\} \subseteq E \). Then, \( E_{\text{ fus } A} \) denotes the set \( \{\{e_1,\ldots,e_k\}, e_{k+1},\ldots,e_n\} \) i.e. in the new set, \( A \) is treated as a single element. Let

\( \Pi = [E_1, E_2, \ldots, E_t] \) be a partition of \( E \). Let \( A \subseteq E_t, t \leq t \). Then \( \Pi_{\text{ fus } A} \) is the partition

\( E_1, \ldots, E_{t-1}, E_{t_{\text{ fus } A}}E_{t+1}, \ldots, E_t \)

of the set \( E_{\text{ fus } A} \). Let \( G \) be a graph and let \( A \subseteq V(G) \) (where \( V(G) \) is the set of vertices of \( G \)). Then \( G_{\text{ fus } A} \) is the graph on the set of vertices \( V(G)_{\text{ fus } A} \) and edges \( E(G) \) obtained by treating all the nodes of \( A \) in \( G \) as a single supernode.

We now state the following simple Lemma without proof.

Lemma 1

Let \( B \) be a bipartite graph \( (V_1, V_2, E(B)) \). Let \( \Pi \) be a partition of \( V \), which minimises \( \{L_0(\cdot) - \lambda\} \). Let \( A \) be a subset of a block of \( \Pi \). Let \( B' = B_{\text{ fus } A} \). Then \( \Pi_{\text{ fus } A} \) minimises \( \{L_0(\cdot) - \lambda\} \).

It is clear from Lemma 1 that if somehow we detect a nonsingleton, nonvoid set \( A \) contained in a block of some partition \( \Pi \) that minimises \( (L_0(\cdot) - \lambda) \) we can reduce the
size of the problem by working with $B_{\text{ fus}}A$. Such a set is described in the Lemma below.

**Lemma 2**

Let $A \subseteq V$, satisfy the following properties.

1. $L(A) - \lambda < \sum_{e \in A} L(e) - \lambda | A |$.
2. If $D \subseteq A$ then $L(A) - L(D) \leq \sum_{e \in (A-D)} L(e) - \lambda | A - D |$.
3. The collection of all subsets of $A$ which satisfy property (1) have a common element.

Then there exists a partition $\Pi$ that minimises $L(\cdot) - \lambda$ such that $A$ is contained in some block of $\Pi$.

The problem of detecting a set which satisfies the properties in Lemma 2 can be converted into one of repeated flow maximisation in a flow graph derived from $B$.

The algorithm for minimising $L(\cdot) - \lambda$, by repeatedly 'fusing' sets which satisfy the properties of Lemma 2 until no such sets exist, has complexity $O(E^2V_2)$.

Next we consider the problem of finding all the critical values of $L(\cdot)$ and the Principal Sequence of partitions.

**Lemma 3**

Let $\Pi_1$, $\Pi_2$ minimise $(L(\cdot) - \lambda_1)$, $(L(\cdot) - \lambda_2)$ respectively with $\lambda_1 \geq \lambda_2$. Then

(a) $\Pi_2 \geq \Pi_1$.
(b) If $\Pi_1 \neq \Pi_2$, $(L(\cdot) - \lambda_0)(\Pi_1) = (L(\cdot) - \lambda_0)(\Pi_2)$, and $\Pi_1, \Pi_2$ minimise $(L(\cdot) - \lambda_0)$ then $\lambda_0$ is a critical value of $L(\cdot)$.

Lemma 3 suggests the following simple procedure for detecting all the critical values of $L(\cdot)$.

Find $\lambda_0$ such that $(L(\cdot) - \lambda_0)(\Pi_1) = (L(\cdot) - \lambda_0)(\Pi_2)$. Minimise $(L(\cdot) - \lambda_0)$ finding a partition, say, $\Pi_1$. If $(L(\cdot) - \lambda_0)(\Pi_1) = (L(\cdot) - \lambda_0)(\Pi_2)$ we stop with $\lambda_0$ as the only critical value. Otherwise, we repeat the process with $\Pi_1, \Pi_2$ and with $\Pi_1, \Pi_0$. Finally we are left with a set of partitions $\Pi_0, \Pi_1, \ldots, \Pi_{n-1}$ such that $\Pi_0, \Pi_1$ minimise $(L(\cdot) - \lambda_0)$, $\Pi_1, \Pi_2$ minimise $(L(\cdot) - \lambda_0)$ and so on. If from this sequence we omit repetitions of $\lambda$'s and shorten the sequence of partitions correspondingly, we get the Critical Values and the Principal Sequence of Partitions. The complexity of the algorithm for finding the Principal Sequence of Partitions is $O(E^2V_2)$.

It is quite possible that the algorithm for PLP can result in a redundant Principal Sequence $\Pi_0, \Pi_1$, with a single critical value. In this case the underlying Bipartite graph can be shown to have some good properties.

### 6 Conclusion

In this paper we have shown that the PLA partitioning problem can be regarded as that of partitioning one side of a bipartite graph in such a way that the other side is broken up into sets whose 'overlap' is a minimum. A natural solution to this problem is afforded by constructing the Principal Sequence of Partitions of the function $L(\cdot)$.

**References**

Fig 2(a). PLA to be partitioned

Fig 2(b). Partition on set $V_o$ (Minterms)

Fig 2(c). Partition on set $V_a$ (Primary Inputs only)

Fig 3. An example for PLP.