Optimal datapath allocation for multiple-wordlength systems

G.A. Constantinides, P.Y.K. Cheung and W. Luk

High-level synthesis for multiple-wordlength systems is examined. A formulation of the combined scheduling, binding, and wordlength selection problem is proposed. Integer linear programming is used to obtain area-optimal scheduling, binding and wordlength selection for such systems.

Introduction: There has been significant recent research into multiple-wordlength or multiple-precision systems, where datapaths are constructed from operators with different bit-width [1 – 3]. However, little research has been conducted [3, 4] into high-level synthesis for these systems. The use of multiple wordlengths has a significant impact on the traditional problems of high-level synthesis: scheduling, resource binding, and module selection [5]. This is the result of two factors. First, each computational unit of a specific type, for example 'multiplier', cannot be assumed to have an equal cost in a multiple-precision system [4]. Secondly, the choice of wordlength for an operation can impact on the latency of that operation. The existence of multiple wordlengths therefore complicates the resource binding problem, and also increases the interaction between operation binding and scheduling.

To our knowledge, this Letter is the first formulation of the combined scheduling. resource binding and wordlength selection problem. The formulation is given in terms of an integer linear program (ILP), the solution of which yields optimal results with respect to the cost function. It is demonstrated that the area-based cost function can be used to provide bounds on the number of resource instances of each size and type, and an example is given.

The notation $f(X)$ for the range of a function f: $X \rightarrow Y$ is used in this Letter. $|X|$ represents the cardinality of set X, and $\wedge$ denotes logical AND.

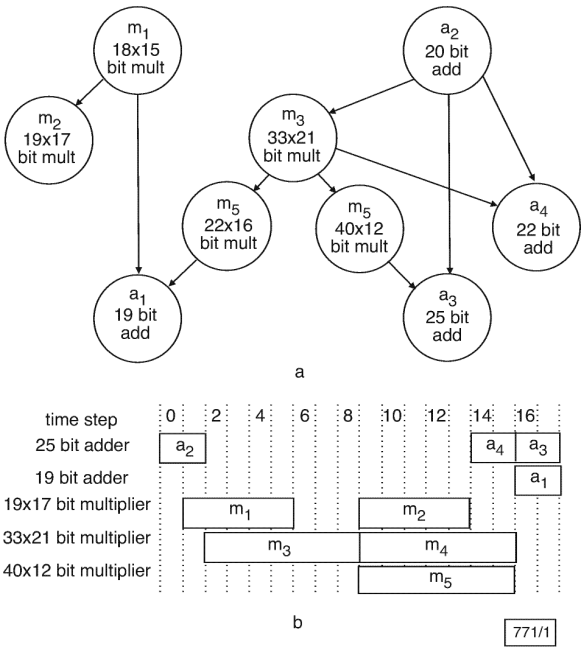

Fig. 1 *Multiple-wordlength sequencing graph and its scheduling, resource binding and wordlength selection*

a Sequencing graph
b Scheduling, resource binding and wordlength selection

Resources and instances: The starting point for our formulation is a sequencing graph $G(O, E)$ [5] and a target latency constraint λ. Define the set of multipliers M and the set of adders A, which together constitute the set of operations $O = M \cup A$. The width of a multiplication operation $m \in M$ is given by a tuple of integers $b^M(m) = (p, q)$, indicating a p-by-q bit multiplier and ordered such that $p \geq q$. Similarly the width of an addition operation $a \in A$ is given by an integer $b^A(a) = p$, indicating a p-bit addition.

There are several resource types which can arise from the sharing of resources between these operations. Specifically, there is a set of adder types $R^A(a)$ that can implement an addition $a \in A$ (eqn. 1) and a set of multiplier types $R^M(m)$ that can implement a multiplication $m \in M$ (eqn. 2). Each resource type is defined by its width. Together, these resource types form a resource-set $R(o)$ for each operation $o \in O$ (eqn. 3).

$$R^A(a) = \{p \,|\, \exists p \in b^A(A) : p \geq b^A(a)\} \tag{1}$$

$$R^M(m) = \{(p,q) \,|\, \exists (p,b) \in b^M(M), \exists (c,q) \in b^M(M) : p \geq c \wedge q \geq b \wedge p \geq d \wedge q \geq e \text{ where } (d,e) = b^M(M)\} \tag{2}$$

$$R(o) = \begin{cases} R^A(o) & o \in A \\ R^M(o) & o \in M \end{cases} \tag{3}$$

Although each resource type may be large enough to cover many different operations, the number of instances of each resource type is limited by the optimisation goal of minimum area. Whenever a set of operations is bound to a single resource, the optimal wordlengths of the resource are large enough to perform all operations bound, but no larger. Thus we can define tight bounds on the largest number of instances $I(r)$ of each type of resource $r \in R(O)$ (eqns. 4 and 5). For an adder resource, there can be as many instances as there are add operations of the resource size. For a multiplier resource, each p-by-q bit resource can only arise due to resource sharing of a p-by-b and a c-by-q resource (where $p \geq c$ and $q \geq b$). The number of these pairings is bounded by eqn. 5.

$$I(r) = |\{a \,|\, a \in A : b^A(a) = r\}| \quad r \in R^A(A) \tag{4}$$

$$I(p,q) = \min\{|\{m | m \in M : p = d \wedge q \geq e \text{ where } (d,e) = b^M(m)\}|, |\{m | m \in M : q = e \wedge p \geq d \text{ where } (d,e) = b^M(m)\}|\} \quad (p,q) \in R^M(M) \tag{5}$$

We define the latency $L(r)$ of resource $r \in R(O)$ according to implementation-specific empirical formulas, which for our implementation on Sonic [6] are defined in eqn. 6.

$$L(r) = \begin{cases} \lceil (p+q)/8 \rceil & (p,q) = r \in R^M(M) \\ 2 & r \in R^A(A) \end{cases} \tag{6}$$

We now calculate the maximum $L_{max}(o)$ and minimum $L_{min}(o)$ latency of each operation $o \in O$ according to eqns. 7 and 8.

$$L_{min}(o) = \min_{r \in R(o)} L(r) \tag{7}$$

$$L_{max}(o) = \max_{r \in R(o)} L(r) \tag{8}$$

Each operation $o \in O$, executing on resource type $r \in R(o)$, can start its execution during any time step in the set $T(o, r)$ (eqn. 9), where Z_+ denotes the set of non-negative integers. To define this, we utilise modified concepts of as soon as possible (ASAP) and as late as possible (ALAP) scheduling. Specifically, we define $ASAP(o)$ to be the time-step of the as soon as possible scheduling of operation o when all operations $o' \in O$ have latency $L_{min}(o')$ (eqn. 7). We also define $ALAP(o, \lambda)$ to be the time-step of the as late as possible scheduling of operation o, given a total of λ time steps, under the same latency condition.

$$T(o,r) = \{t \,|\, t \in Z_+ : t \geq ASAP(o) \wedge t \leq ALAP(o,\lambda) - L(r) + L_{min}(o)\} \tag{9}$$

From this, we define all possible start times $T(o)$ for each operation $o \in O$, according to eqn. 10, and the complete set of time-steps T (eqn. 11).

$$T(o) = \{t \,|\, \exists r \in R(o) | t \in T(o,r)\} \tag{10}$$

$$T = \{t \,|\, \exists o \in O : t \in T(o)\} \tag{11}$$

Finally, we define the area cost for each resource type, an empirical function which for our implementation is given in eqn. 12. Here α is a numerical constant indicating the relative area-consumption of addition and multiplication.

$$cost(r) = \begin{cases} pq & (p,q) = r \in R^M(M) \\ \alpha r & r \in R^A(A) \end{cases} \tag{12}$$

ILP formulation: Extending the notation used by Landwehr *et al*. [7], we formulate the ILP as follows. Define $b_{i,r}$ as in eqn. 13, where for a resource instance to be 'bound' means that at least one operation is bound to it. This allows the optimisation problem to be formulated in eqn. 14.

$$b_{i,r} = \begin{cases} 1 & \text{if instance } i \text{ of resource type } r \text{ is bound} \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

$$\text{Minimise} \sum_{r \in R(O)} cost(r) \sum_{i=1}^{I(r)} b_{i,r} \tag{14}$$

To introduce the constraints, let $x_{o,t,i,r}$ be defined as in eqn. 15.

$$x_{o,t,i,r} = \begin{cases} 1 & \text{if operation } o \text{ is scheduled at time-step } t \\ & \text{on the } i\text{th instance of resource type } r \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

The minimisation is performed subject to four types of constraint. The first comprises the binding constraints, to ensure that each operation is executed on exactly one instance of one resource type (eqn. 16). The second comprises the resource constraints, to ensure that no instance of a resource type is executing more than one operation at a time (eqn. 17). The final set is made up of the precedence constraints, to ensure that all operations obey the dependencies in the sequencing graph (eqn. 18).

$$\forall o \in O, \quad \sum_{r \in R(o)} \sum_{i=1}^{I(r)} \sum_{t \in T(o,r)} x_{o,t,i,r} = 1 \tag{16}$$

$$\forall t \in T, \forall r \in R(O), \quad \forall i \in \{1, ..., I(r)\}$$
$$\sum_{o \in O: r \in R(o)} \sum_{t_1 \in \{t, ..., t+L(r)-1\} \cap T(o,r)} x_{o,t_1,i,r} \leq b_{i,r} \tag{17}$$

$$\forall (o_1, o_2) \in E, \quad \forall t \in T(o_2) \cap \{ASAP(o_1)+L_{min}(o_1)-1, \ldots, ALAP(o_1)+L_{max}(o_1)-1\}$$
$$\sum_{r \in R(o_2)} \sum_{i=1}^{I(r)} \sum_{t_2 \in T(o_2,r): t_2 \leq t} x_{o_2,t_2,i,r} + \sum_{r \in R(o_1)} \sum_{i=1}^{I(r)} \sum_{t_1 \in T(o_1,r): t_1 > t-L(r)} x_{o_1,t_1,i,r} \tag{18}$$

Example and conclusion: Fig. 1 illustrates a simple example. Fig. 1*a* is a sequencing graph $G(O, E)$ annotated with multiple-wordlength information. The ILP formulation contains 164 variables and 166 constraints for $\lambda = 18$, the critical path length. Fig. 1*b* illustrates an optimal solution corresponding to this λ, with the following variables taking the value 1: $x_{a_2,0,1,25}$, $x_{a_4,14,1,25}$, $x_{a_3,16,1,25}$, $x_{a_1,16,1,19}$, $x_{m_1,1,1,(19,17)}$, $x_{m_2,9,1,(19,17)}$, $x_{m_3,2,1,(33,21)}$, $x_{m_4,9,1,(33,21)}$, $x_{m_5,9,1,(40,12)}$, $b_{1,25}$, $b_{1,19}$, $b_{1,(19,17)}$, $b_{1,(33,21)}$, $b_{1,(40,12)}$. All other variables are equal to zero.

An ILP formulation of the datapath allocation and scheduling problem, suitable for multiple wordlength systems, has been presented. It has been shown that tight bounds on the number of variables and constraints in the ILP formulation can be found by utilising the nature of the cost-function to decide on the number of instances of each resource type necessary. Resource-types are automatically extracted from the input sequencing graph.

Optimal solutions can only be found for relatively small examples using ILP due to the large number of variables and constraints. Our current research is focusing on efficient heuristic solutions to this problem and incorporating this synthesis technique within the Synoptix wordlength optimisation system [1].

23 June 2000
Electronics Letters Online No: 20001044
DOI: 10.1049/el:20001044

G.A. Constantinides and P.Y.K. Cheung (*Electrical and Electronic Engineering Department, Imperial College, London SW7 2BT, United Kingdom*)

W. Luk (*Department of Computing, Imperial College, London SW7 2BZ, United Kingdom*)

References

1 CONSTANTINIDES, G.A., *et al.*: 'Multiple precision for resource minimization'. Proc. IEEE Symp. Field-Programmable Custom Computing Machines, Napa, CA, April 2000

2 CMAR, R., *et al.*: 'A methodology and design environment for DSP ASIC fixed point refinement'. Proc. Design Automation and Test in Europe, Munich, Germany, March 1999, pp. 271–276

3 KUM, N., and SUNG, W.: 'Wordlength optmization for high-level synthesis of digital signal processing systems'. Proc. IEEE Int. Workshop Signal Processing Systems, October 1998, pp. 569–578

4 CONSTANTINIDES, G.A., *et al.*: 'Multiple-wordlength resource binding' *in* GRUENBACHER, H., and HARTENSTEIN, R. (Eds.): 'Field-programmable logic: The roadmap to reconfigurable systems' (Springer-Verlag, Berlin, 2000)

5 DEMICHELI, G.: 'Synthesis and optimization of digital circuits' (McGraw-Hill, New York, 1994)

6 HAYNES, S.D., *et al.*: 'Video image processing with the Sonic architecture', *IEEE Computer*, 2000, **33**, (4), pp. 50–57

7 LANDWEHR, B., *et al.*: 'OSCAR: Optimum simultaneous scheduling, allocation and resource binding based on integer programming'. Proc. European Design Automation Conf., Grenoble, France, September 1994, pp. 90–95