A Novel Algorithm for Sinking the Access of Memory in a Processor

P.N.PALANISAMY [1], S.MEERA [2], K.JAYANTHI[3], S.MANOJKUMAR[4]

[1, 2, 3, 4] *ASSISTANT PROFESSOR*
DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
MAHENDRA COLLEGE OF ENGINEERING, SALEM

ABSTRACT

Reducing memory access in embedded digital signal processing application is a need for efficient use of energy. The power is a leading constraint factor to reduce and achieve of low power signal processing. In proposed technique the dsp stone processor is taken as benchmark circuit and it has been processed. By implementing memory mapping technique and use of matrix method the registers has been taken into account and reducing number of loops. The reduction of loops in processor is leads to reduce number of memory access and it achieves low power.

Keywords: ***DSP, Loop unrolling, REALM, Embedded systems.***

1. INTRODUCTION

An embedded system is a computer system designed for specific control functions within a larger system, often with real-time computing constraints. Embedded processors can be classified into two broad categories. Ordinary microprocessors include separate integrated circuits for memory and peripherals. Digital Signal Processor is special processors with their processing techniques which is set suit the Signal Processing Applications. MAC (Multiply and Accumulate) and Shifter (Arithmetic and Logical shift) units are added to the DSP cores hence Signal Processing Algorithms are depend on MAC and shifter operations. Circular Buffers, Bit Reversal Addressing, Hardware Loops, and DAG are special features of a DSP Architecture.

Memories are playing major role in embedded systems. In embedded digital signal processing applications memory access utilizes the more energy. Therefore reducing memory access minimizes the consumption of energy.

2. RELATED WORK

i) SOFTWARE OVERLAYS

Low cost micro-processor generally do not have an in-built cache controller. But on these devices it may be still desirable to keep the currently being used code in internal memory and replace it with a new code section when it is not

being used. This can be done using "Software Overlays".

ii) ARRAY PADDING

The array padding transformation is designed in order to set a new size of an array and the main aim of this transformation is to reduce the number of conflicts memory in a system. The major tasks that affect the memory accesses are scheduling of memory accesses in code, allocation of memory modules, and binding variables to memories. In embedded systems, particularly those with high data computations, the delay of memory access is one of the major bottlenecks in the system's performance.

It solves determination of memories, mapping of arrays to memories, Scheduling of memory-access operations.

iii) SCALAR REPLACEMENT

It has been shown that the allocation of array values to registers can significantly improve the memory performance. However, in many product compilers array references are left as references to main memory rather than promoted to references to registers because the data-flow analysis done by the compiler is not powerful which is not recognize most opportunities for reuse in the subscripted variables. Arrays are often treated in a particularly naive fashion, if at all, making it impossible to determine when a specific element might be reused.

3. PARTIAL LOOP PARTIONING

In compiler PRE is a compiler optimization that eliminates expressions that are redundant on some but not necessarily all paths through a program. PRE is a form of common sub expression elimination.

i) LOOP SCHELDULING FOR HIDING MEMORY

In loop scheduling method there are ALU schedule for the ALU computations, first-level schedule for the first-level memory operations, second-level schedule for the second-level memory operations. The longest of these three schedules determines the overall system schedule length. Figure 1 represents the loop scheduling for hiding memory. It represents the memory mapping and ALU computations for the memory operations and it schedules the system modelling. In embedded systems, particularly those with high data computations, the delay of memory access is one of the major bottlenecks in the system's performance Memories are playing major role in embedded systems. In embedded digital signal processing applications memory access utilizes the more energy. Therefore

reducing memory access minimizes the consumption of energy.

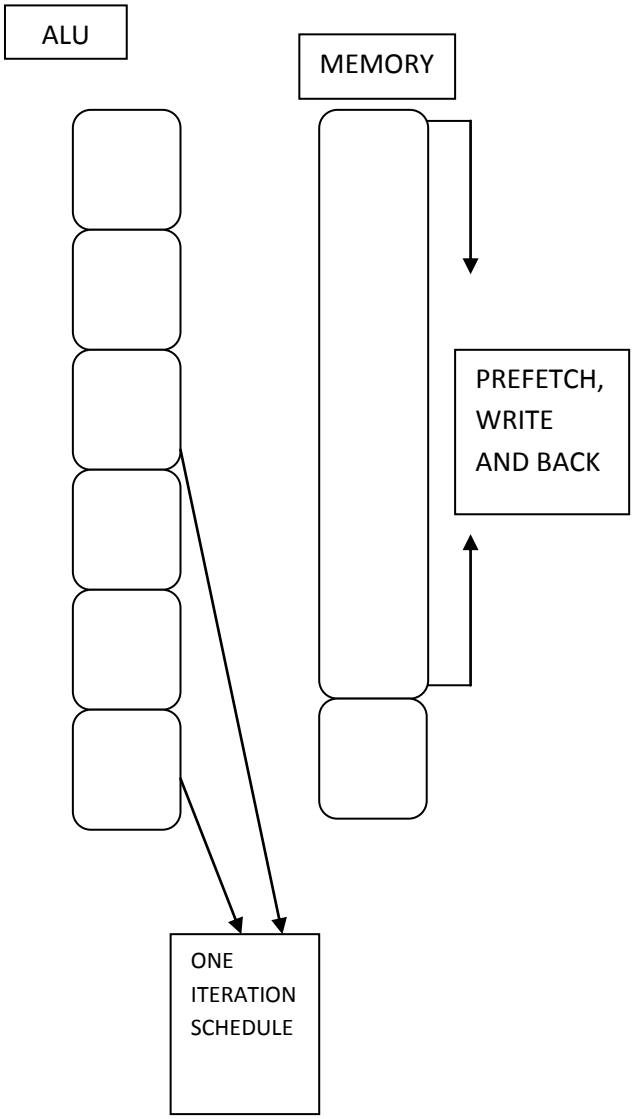

Fig 1: loop scheduling for hiding memory

ii) VARIABLE PARTITIONING FOR DSP PROCESSORS

A graph model known as the variable independence graph (VIG) and algorithms is used to tackle the variable partitioning problem.VIG is more effective than interference graph for solving variable partitioning problem. Then, a scheduling algorithm known as the rotation scheduling with variable repartition (RSVR) to improve the schedule lengths efficiently on multiple memory module architecture. The scheduling algorithm which adjusts the partitions of variables during scheduling and generates a compact schedule. The compact schedule is based on retiming and software pipelining.

iii) LOOP UNROLLING

Loop unrolling is the technique of reusing the loop code which includes more than one iteration of the old code, in a single pass in addition with the new one. Loop unrolling is performing by replicating the inner loop by some number of times and scheduling the performance code as a single basic block.

Loop unrolling has two advantages

1. Producing a larger loop body provides a larger block of instructions for the scheduler to work with which gives the scheduler more options when positioning operations.
2. Combining multiple operations allows induction variable computation to be combined.

The performance improvements are pointed towards the potential penalty which occurs by increased I-cache misses on the larger loop body. Loop unrolling

technique is used to minimize stalls that may be encountered inside loops, and also to get rid of the overhead of running unnecessary branch conditionals. To keep pipeline full, parallelism among instruction must be exploited by finding sequences of unrelated instructions that can be overlapped in the pipeline. To avoid stalls, a dependent instruction must be separated from the source instruction by a distance in clock cycles equal to the pipeline latency of that source instruction. Loop unrolling uses memory mapping technique and acquire memory from processor then by using matrix method of code it tends to reduce number of loops in memory. So by reducing number of memory cycles it makes to reduce number of memory access in processor and also it leads to achieve low power in signal processing. The number of loops is depends upon the number of load and store operation used in memory architectures and registers.

4. Matlab code for loop unrolling

Unrolling 1

```
clc
clear all
close all
A=zeros(1,100);
C=zeros(1,100);
A(1)=5;
A(2)=5;
C(1)=2;
C(2)=2;
delay=1;
tic;
for i=2:delay:60-delay
a=C(i-1);
   b=C(i);
   c=a+b;
   A(i+1)=c;
   d=A(i-1);
   e=d+b;
   C(i+1)=e;
end
toc
(i+1)/delay
disp('no.of loops')
```

number of memory cycles has been detected as 60 loops

Unrolling 2

```
clc
clear all
close all
A=zeros(1,100);
C=zeros(1,100);
A(1)=6;
A(2)=6;
C(1)=5;
C(2)=5;
delay=3;
tic
for i=3:delay:60-delay
a=C(i-1);
   b=C(i);
```

```
    c=a+b;
    A(i+1)=c;
    d=A(i-1);
    e=d+b;
    C(i+1)=e;
    f=b+e;
    A(i+2)=f;
    g=A(i);
    h=g+e;
    C(i+2)=h;
    l=h+g;
    A(i+3)=l;
    k=A(i);
    m=k+h;
    c(i+3)=m;
end
toc
(i+3)/delay
disp('no.of loops')
```

the loop unrolling method the reduces to 20 loops and hence it reduce the memory cycle and achieve low power.

5. Conclusion

Reducing memory accesses are important for embedded digital signal processing application since they need to be executed with low energy consumption. By using loop unrolling memory access reducing techniques, memory accesses can be reduced in embedded digital signal processing application to achieve efficient use of energy. In the loop unrolling the memory cycle has been reduced and low power has been achieved. Further by using REALM technique the number of memory cycle has been reduced and avoid unwanted power use by memories in processor.

REFERENCES

[1] Q. Wang, N. Passos, and E. H.-M. Sha, "Optimal loop scheduling for hiding memory latency based on two level partitioning and prefetching," IEEE Trans. Circuits Syst. II, Analog Signal Process., vol. 44, no. 9, pp. 741–753, Sep. 1997.

[2] Y. Song and Z. Li, "Applying array contraction to a sequence of doall loops," in Proc. Int. Conf. Parallel Process., 2004, pp. 46–53.

[3] Y. Ding and Z. Li, "A compiler scheme for reusing intermediate computation results," in Proc. Ann. IEEE/ACM Int. Symp. Code Generation Opt. (CGO), 2004, pp. 279–291.

[4] Z.Wang, E.H-M Sha, and x.S.Hu, "Combined portioning and data padding for scheduling multiple loop nests," in Proc. International conference compilers Arch.,sunth. Embed system 2001,pp. 67-75

[5] D.Callahan, S.Carr, and K.kennedy, "improving register allocation for subscriped variables," in Proc. ACM SIGPLAIN Conf. program. Lang Des implementation, 1990, pp. 53-65.

[6] V.Sarkar and G.R.Gao,"Optimization of array access by collective loop transformations", in Proc.

5th International Conference Supercomputer, 1991, pp.194-205.

[7] G.R.Gao, "A maximally pipelined tridiagonal linear equation solver," J.Parallel Distrib Comput., vol.3 no.2, pp.215-235, Jun 1986.

[8] L.E.L A.P. Brown and K.K.Parhi, "unfolding and retiming for high level DSP synthesis," in proc International Symposium circuits system 1991, pp.2351-2354.