



Parametric Architecture for Implementing Multimedia Algorithms

N. Kavvadias and S. Nikolaidis

Electronics and Computers Div.,
Department of Physics,
Aristotle University of Thessaloniki,
54124, Thessaloniki, Greece

E-mail: nkavv@skiathos.physics.auth.gr



Introduction







- Multimedia applications are characterized by high computational demands
- Conventional processing approaches often prove ineffective in terms of power and/or performance
- A solution that provides dramatic performance improvements is more than welcome



Multimedia algorithm characteristics

- Popular cases are image and video processing, speech/audio, filtering etc
- Mainly consist of loop structures describing regular, repetitive computations on array data sets
- Algorithm-level transformations (e.g. data-reuse) introduce additional loops and add to algorithm complexity

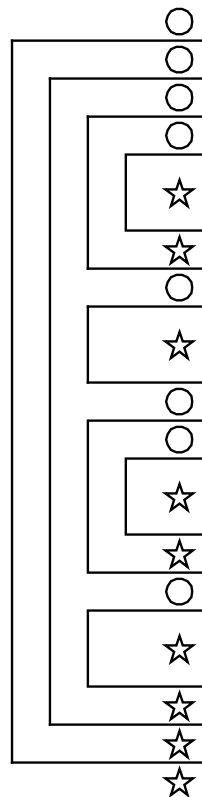
Implementation approaches for multimedia systems

Impact on	Performance	Power	Flexibility
ISP			
ASIC			

- ✓ Embedded ISP: special (loop) cache, hardware loop mechanisms for zero overhead branching
- ✓ New solution: *mapping any number and combination of loops on parametric control hardware (ASIC or FPGA)*

Algorithm flow mapping

- Multimedia algorithm: a loop structure where data processing tasks are positioned

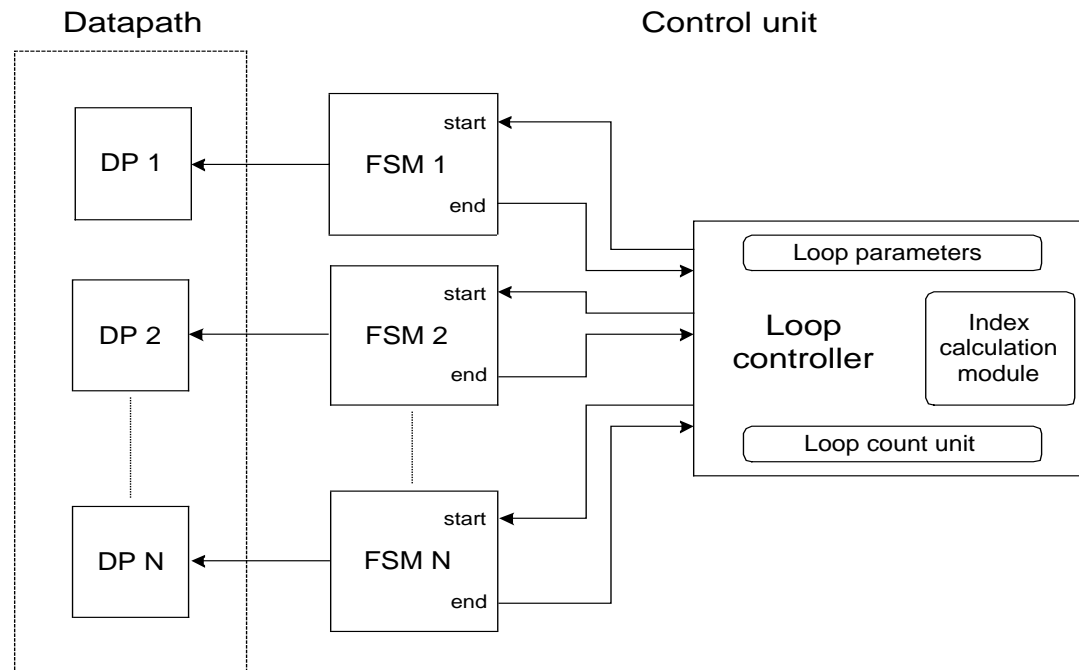


- FOR loop
- forward (fwd) task
- ☆ backward (bwd) task

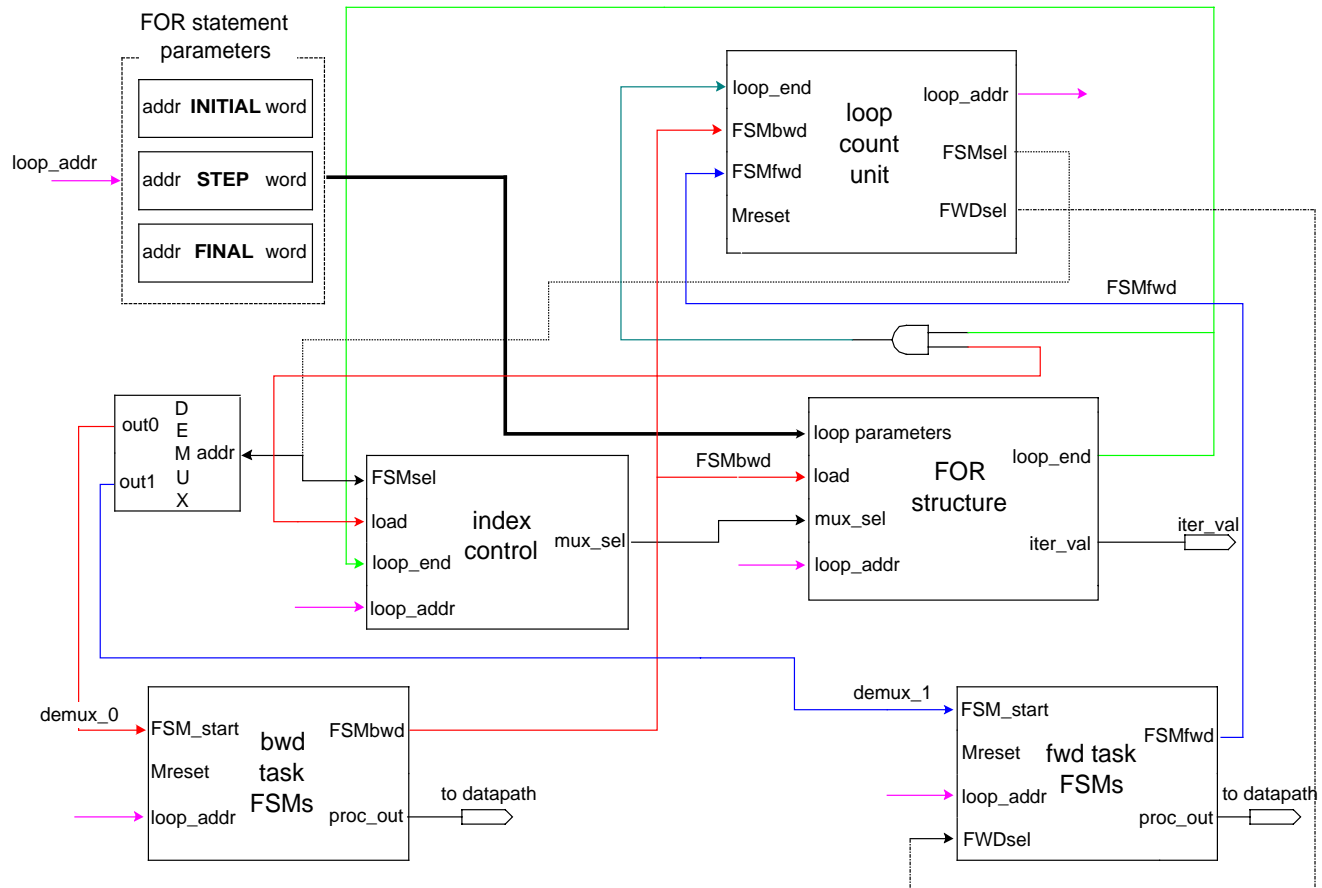
- Tasks realize data processing
 - **bwd** tasks (in *innermost* loops): advance loop indices
 - **fwd** tasks (between *adjacent loops*): no effect on loop indices

Architecture template for a multimedia platform

- The loop controller directs the data processing tasks execution flow so that the appropriate FSM undertakes control of the datapath

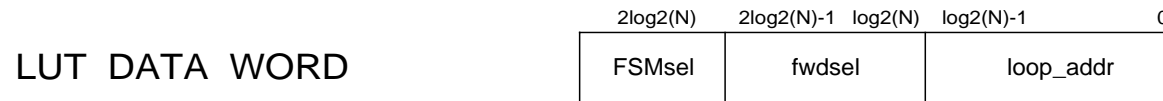
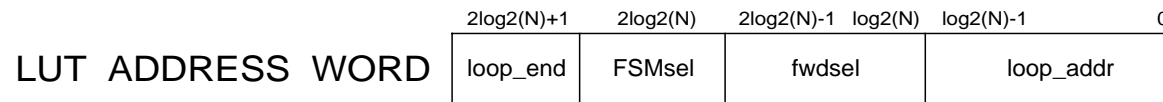


Parametric Control Architecture



LUT address and word formation

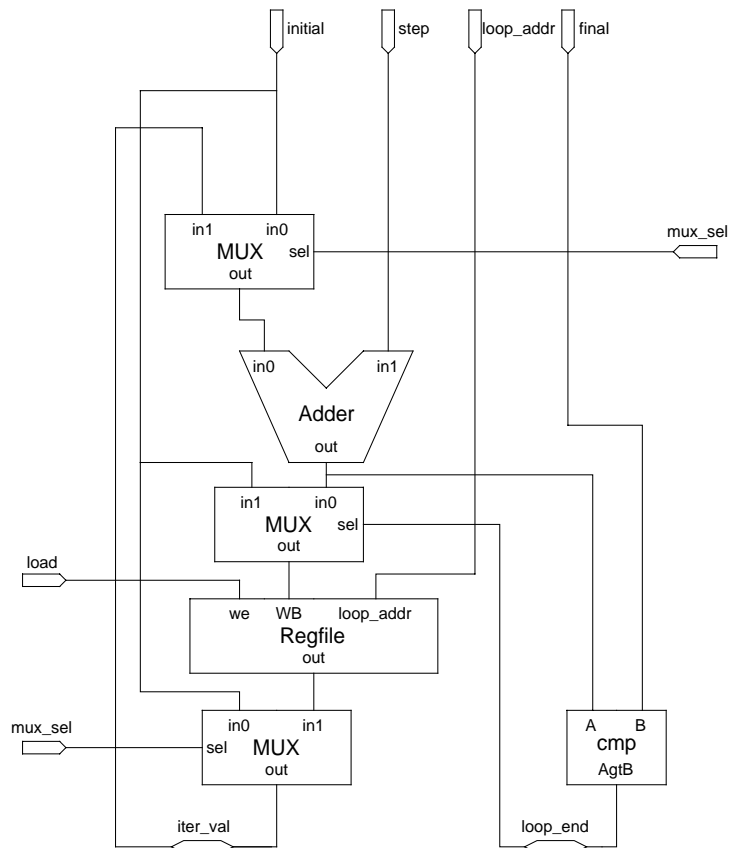
- Loop sequencing information is mapped in a LUT incorporated by the loop count unit



N: maximum number of loops supported

- **loop_addr**: current loop address
- **FSMsel**: task type selection
- **fwdsel**: specific fwd FSM in given loop
- **loop_end**: current loop termination signal

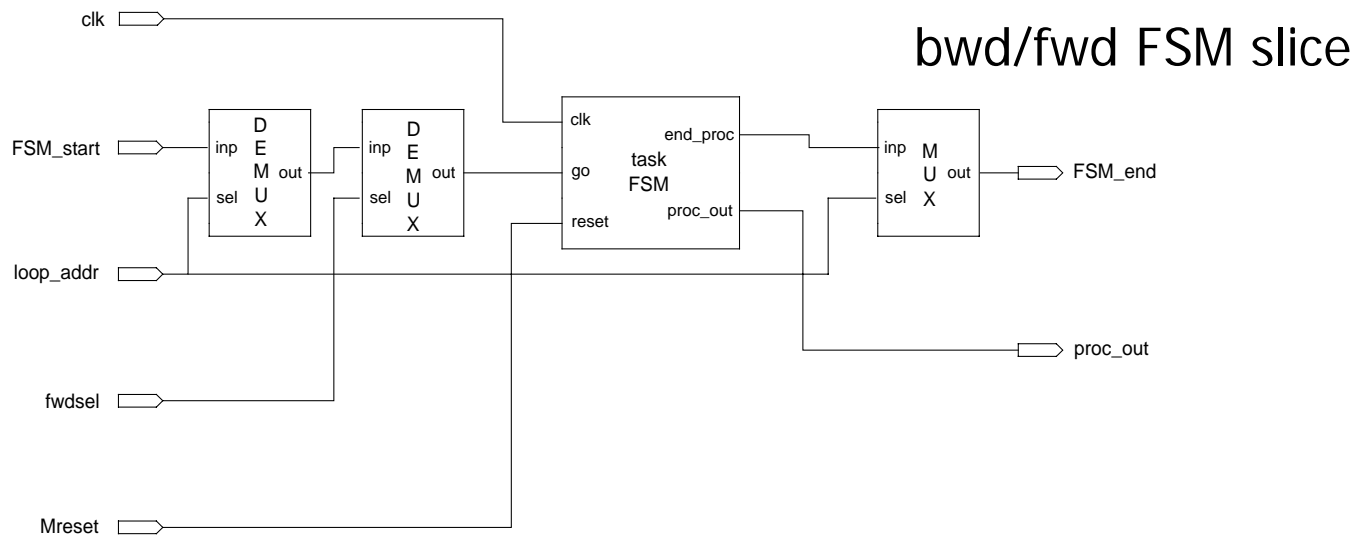
Index calculation module (FOR structure)



- Generates index values
- Current index is calculated by adding step to a previous index value
- Multiplexers for passing initial or stepped values
- A comparison between stepped and final value activates loop termination
- Intermediate indices are stored in a local register file

bwd/ fwd task FSM modules

- Data processing tasks are controlled by their corresponding local FSM



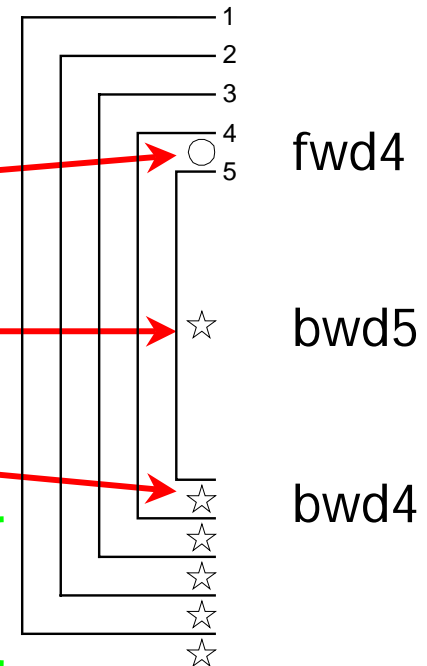
- The appropriate FSM is selected by demultiplexing `loop_addr` (bwd) or `loop_addr` and `fwdsel` (fwd)

Case study of a popular multimedia algorithm

- The blocked matrix multiplication algorithm was evaluated for execution performance

```
for i=0 to H-1, step B
  for j=0 to W-1, step B
    for k=0 to B-1, step 1
      for l=0 to B-1, step 1
        temp = 0
        for m=0 to B-1, step 1
          temp = temp + coeff[k,m]*image_in[i+m,j+l]
        end for
        image_out[i+k,j+l] = temp
      end for
    end for
  end for
end for
```

tasks to close the loops





Evaluation of the method

Algorithm	Proposed architecture	ARM7TDMI	Improvement factor
1 FOR loop iteration	1	6	6
Matrix multiplication	304,000	1,780,000	5.8
Full-Search Motion Estimation	4,000,000	72,700,000	> 18

- Loop intensive algorithms are executed at high performance figures



Conclusions

- An efficient parametric control architecture is presented for implementing multimedia applications
- Fundamentally different approach to assembly-level instruction buffering for a small number of loops
- The proposed architecture is able to execute structured algorithms for any combination of loops
- Loop index update and loop termination are performed in a single clock cycle