A Multi-Objective Integer Linear Program for Memory Assignment in the DSP Domain

G. Grewal, A. Morton, S. Coros, and D. Banerji,

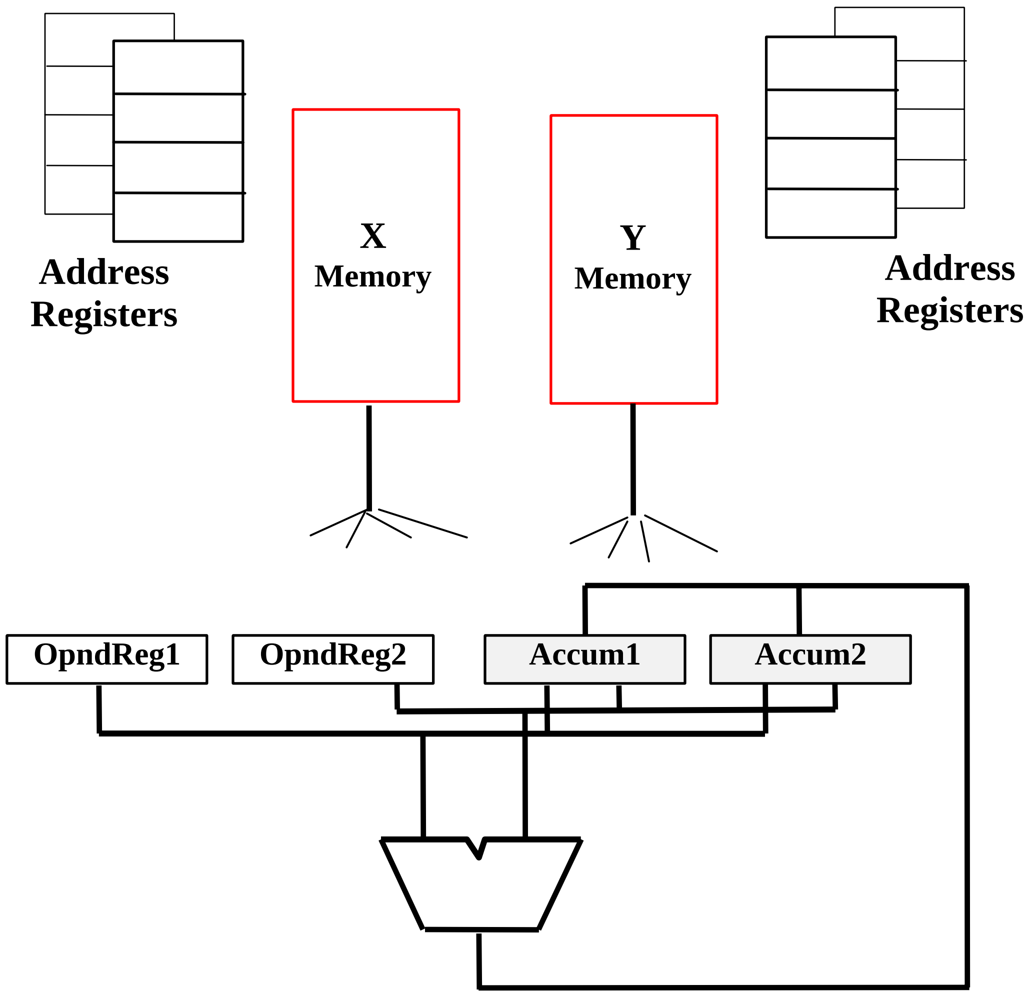
Presented by: Andrew Morton

Dept. of Electrical and Computer Engineering
University of Waterloo
Waterloo, Ontario, Canada

Outline

1. Typical DSP architecture
2. The memory assignment problem for dual memories
3. Constraints and Objective functions
4. Results
5. Closing comments

Typical DSP Architecture

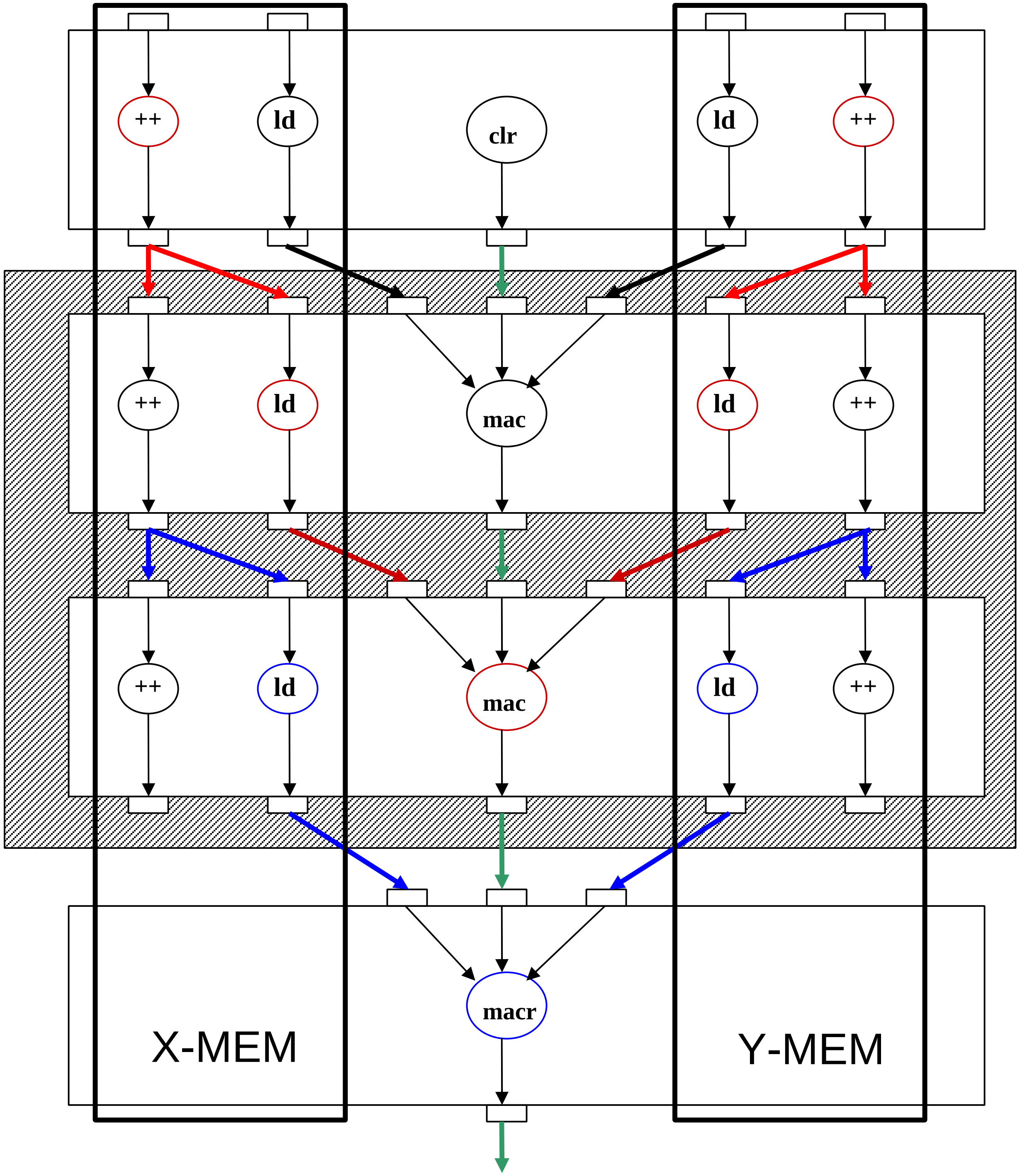


Sample Loop Kernel

```
double FIR_filter(in double A[], in double B[], in int tap) {
    int k;
    double sum=0;

    for(k=0; k< tap; k++)
            sum += A[k] * B[k];
    return sum;
}
```

ALU		X-Memory	Y-Memory	
CLR	A	X:(R0)+, X0	Y:(R4)+,Y0	[1]
REP	#N-1			[2]
MAC	X0,Y0, A	X:(R0)+, X0	Y:(R4)+, Y0	[3]
MACR	X0, Y0, A			[4]



++
ld
clr
ld
++
++
ld
mac
ld
++
++
ld
mac
ld
++
macr
X-MEM
Y-MEM

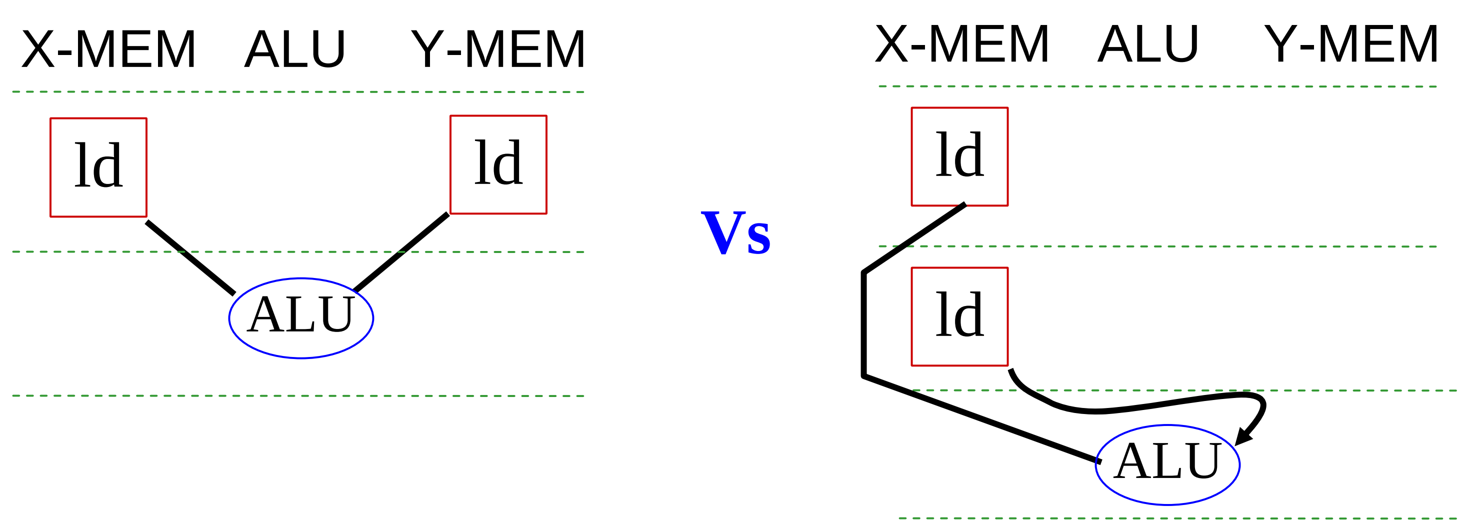
Goal of Data Partitioning

- Goal: To partition the data, prior to scheduling, to create a climate for very parallel, short schedules.

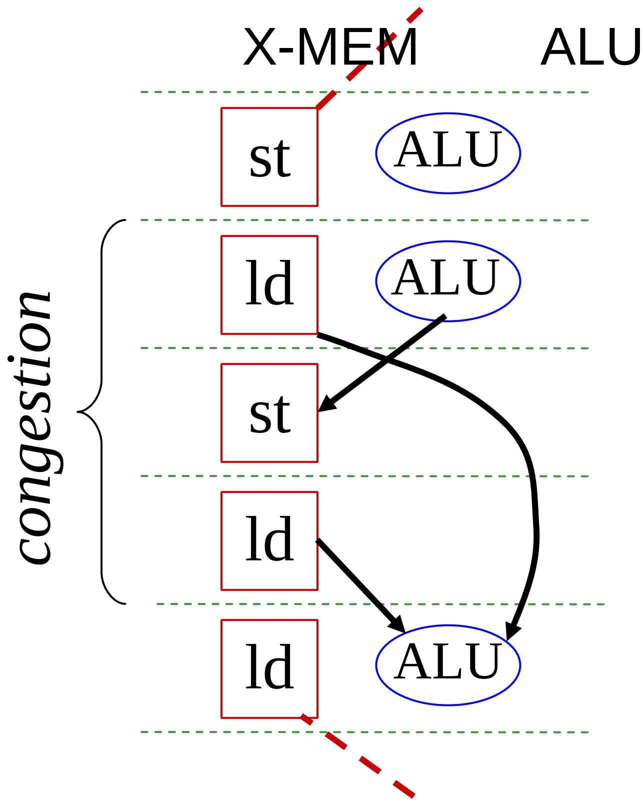
- Problems:
  - Unbalanced Memory Activity
  - Connectivity and non-commutative operations

Load Balancing

- Parallelism in schedules of memory operations promotes the possibility of better schedules.



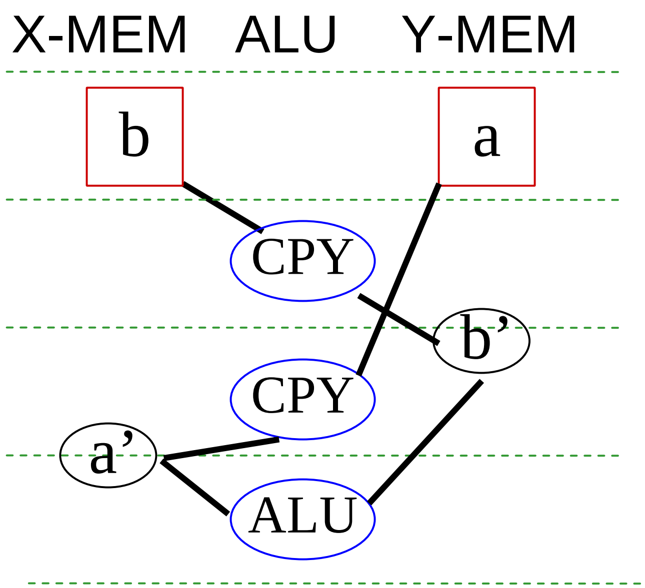
Load Balancing Cont'd



Load balancing can be achieved using:

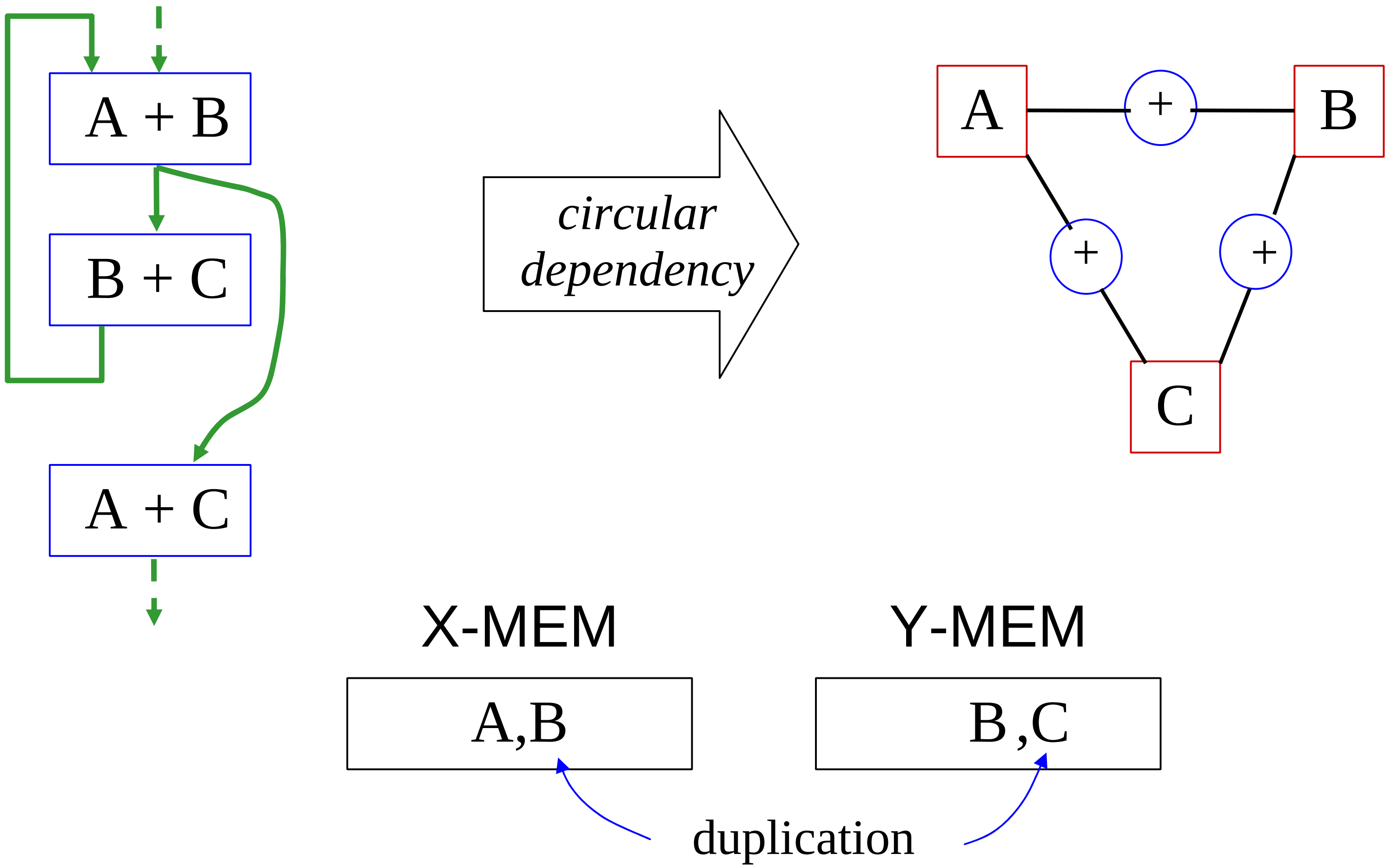
- Objective Function

Register Connectivity



non-commutative

Register Connectivity Cont'd



Remembering the Problem

- Goal: To partition the data, prior to scheduling, to create a climate for very parallel, short schedules.
  - Minimize delays due to values appearing in “wrong” memory
  - While satisfying all *hard* constraints
    - E.g., not exceeding size capacity of either memory

Our Model

- We model the problem as an Integer Linear Program

- The objective function aims to minimize the cost of:
  - Having operands in “wrong” memory – for non-commutative operations
  - Need to update both memories
  - Need to fetch paired operands on separate control steps – for commutative operations

• Block Execution Frequencies used to weight individual objectives can be found through simulation, estimation, experience, etc.

Our Model

- Primary Constraint
  - capacity of either memory cannot be exceeded
- Additional Constraints (Preferences)
  - item *i* must appear in *x* memory
  - item *i* cannot appear in *x* memory
  - items *i* and *j* must have instances in different memories
  - items *i* and *j* must have instances in same memory
  - item *i* cannot be duplicated

Results

- Randomly generate three sets of problems:
  - Set A – 50 variables
  - Set B – 100 variables
  - Set C – 200 variables
- For each set, we varied:
  - # of variables preferred in the left memory (25%-50%)
  - # of variables preferred in the right memory (25%-50%)
  - # of variables that require updating (25%-50%)
  - # of variables that appear in commutative operations (50%-75%)
- Each set contains 16 problem instances

Results

Set	Variables	Solution Times
A	50	0.1s
B	100	1.0 – 2.0 s
C	200	6.0 – 12.0 s

- Lindo solver on Windows 2000, 2.4 GHz P4, 512MB
- Solutions to 5000 variable problems in < 60s

Thanks for Listening