



UNIVERSITAT POLITÈCNICA  
DE CATALUNYA

# A Comparative Study of Modulo Scheduling Techniques

**Josep M. Codina, Josep Llosa and Antonio González**

Dept. of Computer Architecture  
Universitat Politècnica de Catalunya  
Barcelona, SPAIN

**E-mail: {jmcodina, josepl1, antonio}@ac.upc.es**



# Software Pipelining

- ❑ **Instruction Scheduling for VLIW/Superscalar Processors**
  - VLIW processors in DSP market
  - EPIC/IPF
  
- ❑ **Loop Scheduling: Software Pipelining**
  - Loops consume most of the application' execution time
  
- ❑ **Software Pipelining a loop is an NP-complete problem**
  
- ❑ **Software Pipelining big family of techniques**
  - Modulo Scheduling based on heuristics

- ❑ **Modulo Scheduling is an environment to define techniques**
  - Different factors to take into account
  - Lot of techniques can fit in the environment. Different ideas
  
- ❑ **Proposals in the literature evaluated without common**
  - Platform (i.e. compiler)
  - Benchmarks
  - Target architectures
  - Measures
  
- ❑ **Lack of a thorough comparison**

- ❑ **Perform a comparison of state-of-the-art MS techniques**
  - Qualitative
  - Quantitative
  
- ❑ **The work is target to compiler writers**
  - Is one of the techniques better than the others for all architectures?
  - Which is the most powerful technique for a given architecture?



# Talk Outline

- ❑ **Modulo Scheduling Background**
- ❑ **Selection Criteria**
- ❑ **Techniques Compared**
- ❑ **Study Environment**
- ❑ **Results**
- ❑ **Conclusions**

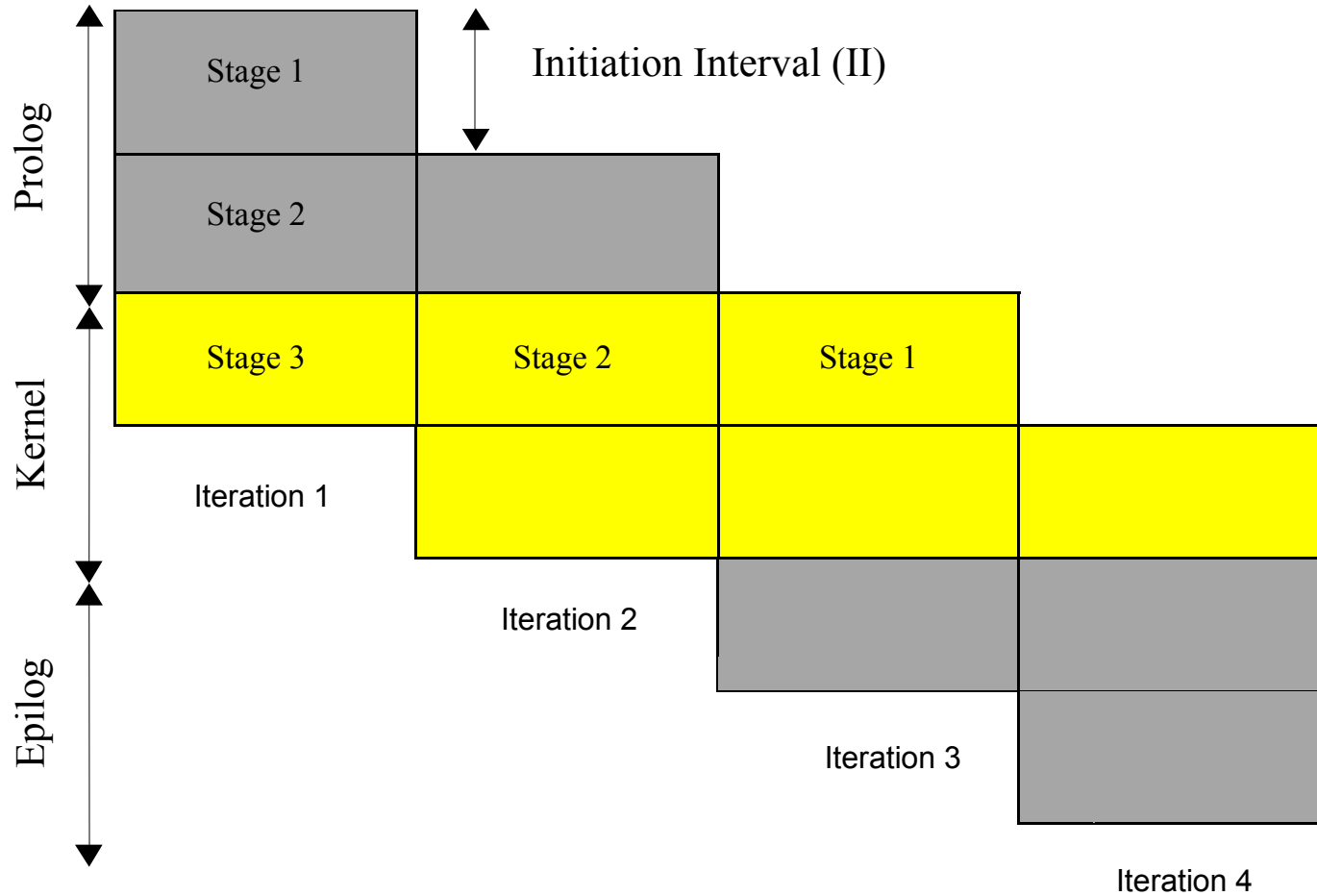


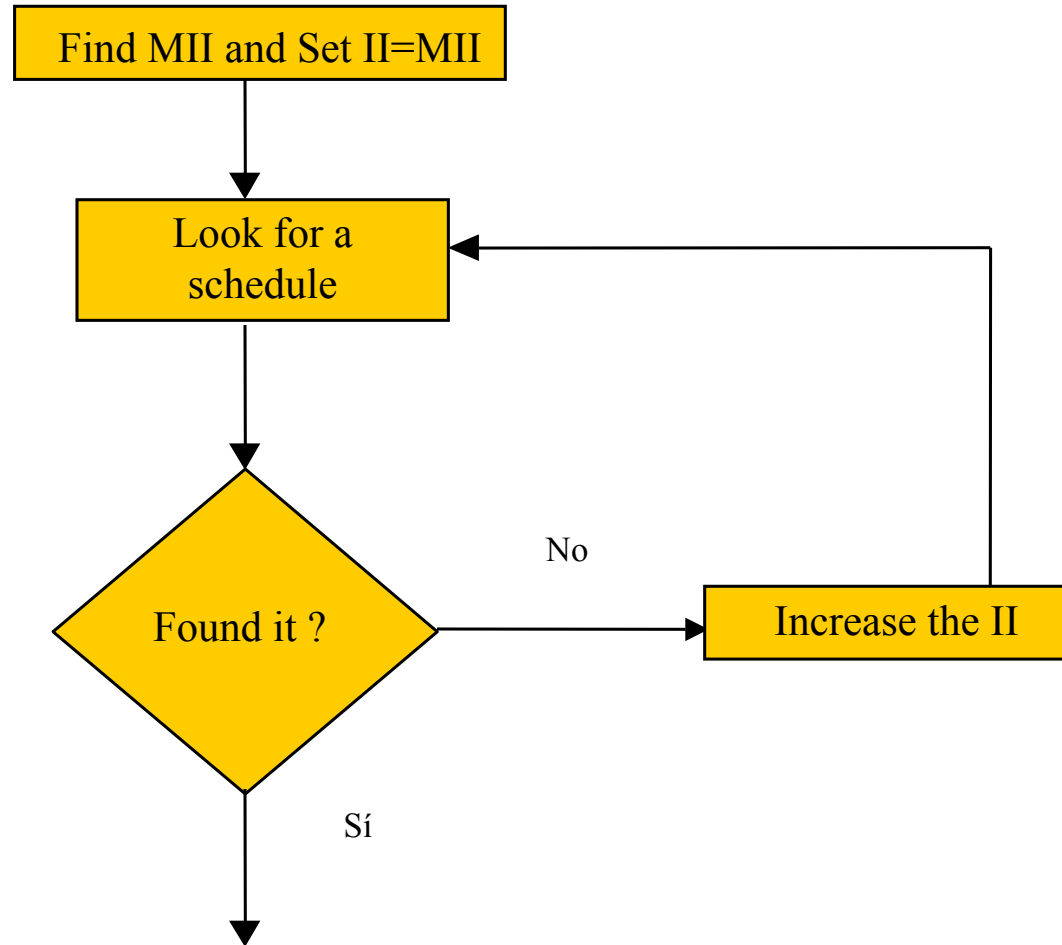
# Talk Outline

- ❑ **Modulo Scheduling Background**
- ❑ Selection Criteria
- ❑ Techniques Compared
- ❑ Study Environment
- ❑ Results
- ❑ Conclusions



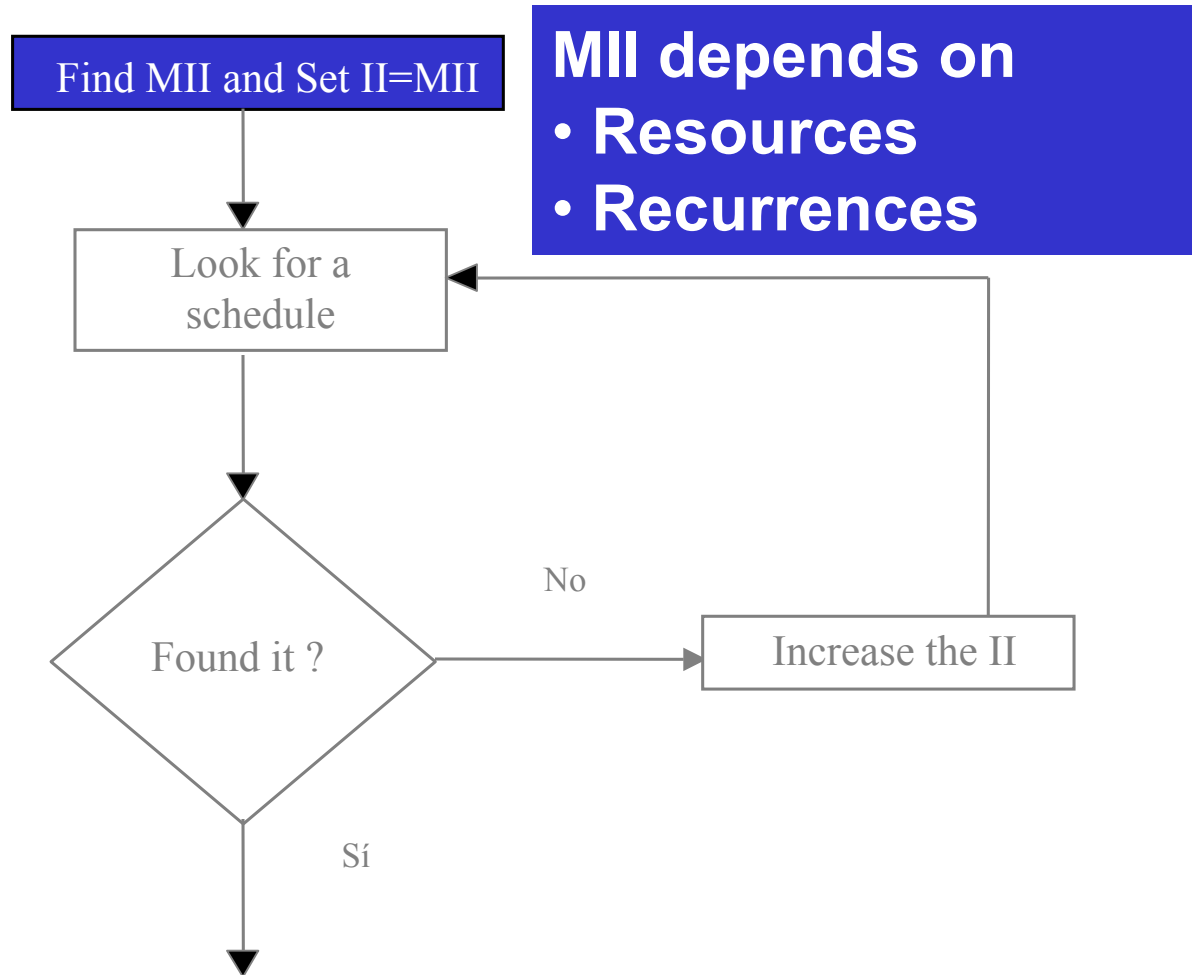
# Basic Ideas

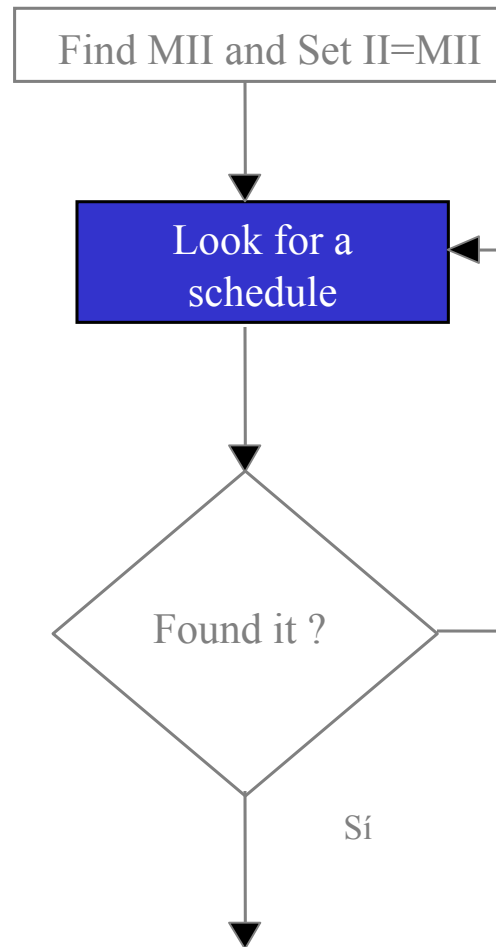






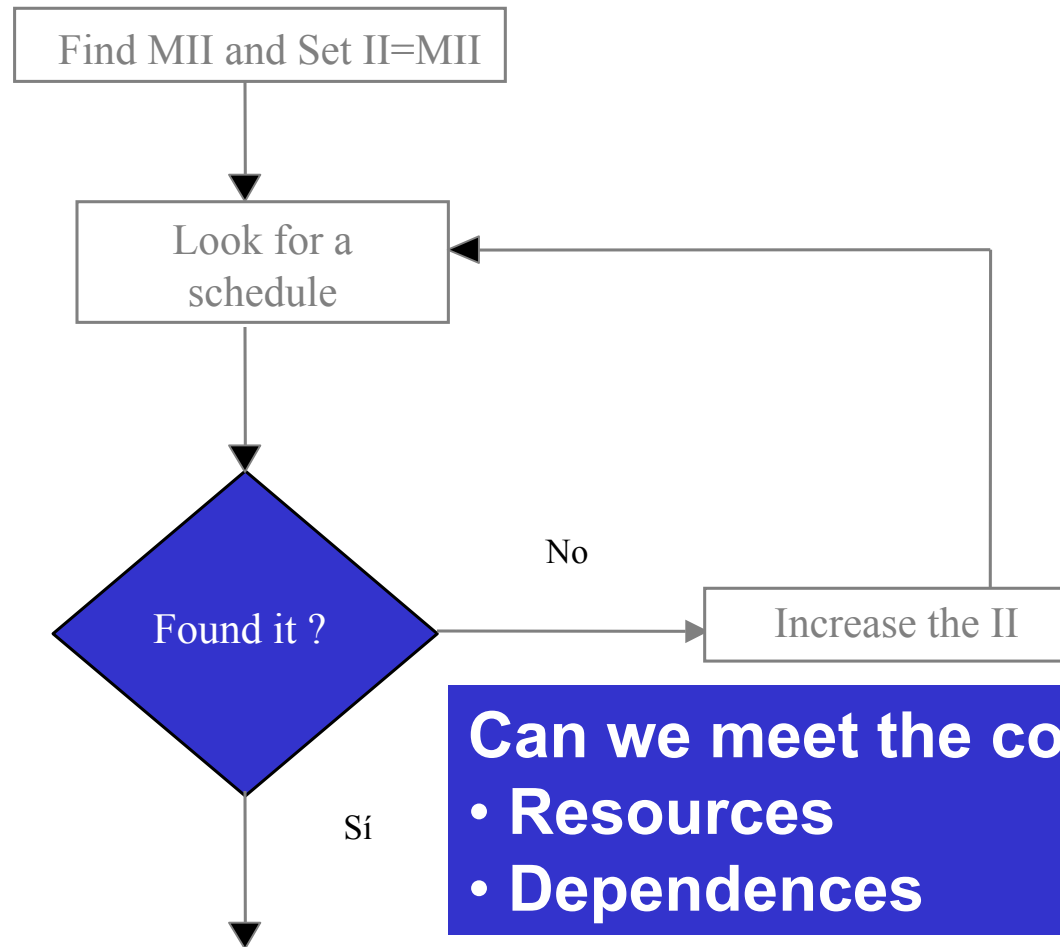
# Basic Scheme





## Look for a schedule

- Ordering the nodes
- Finding a feasible cycle
  - Top-Down/Bottom-up
  - Bi-directional
- When no feasible cycle
  - Use of backtracking
  - Increase the II

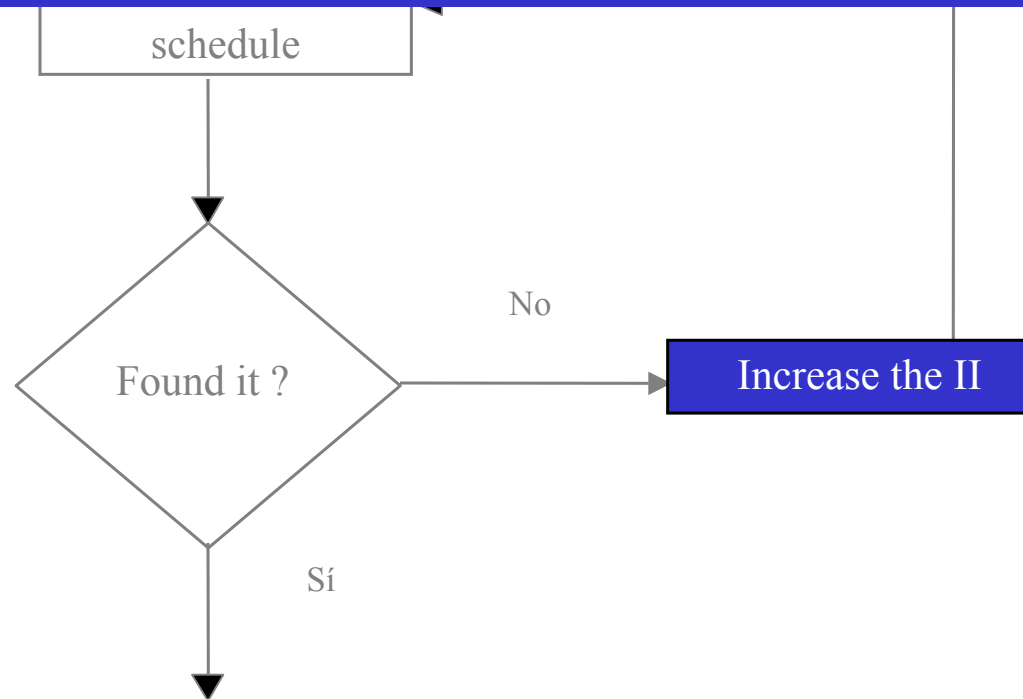


**Can we meet the constraints?**

- Resources
- Dependences

# Basic Scheme

- The larger the  $\Pi$ , the more likely to find a schedule
- The larger the  $\Pi$ , the lower the performance
- $\Pi$  lower than the length of a single iteration



- ❑ **Not always beneficial**
  - Can produce better schedules
  - Can just increase the process of finding a schedule

- ❑ **In some cases, no feasible schedule for a given  $\Pi$**



***Backtracking must be limited***

- ❑ **BudgetRatio:**

Ratio of the maximum number of operation scheduling steps attempted before increasing the  $\Pi$



# Talk Outline

- Modulo Scheduling Background
- **Selection Criteria**
- Techniques Compared
- Study Environment
- Results
- Conclusions



❑ **Value of the code generated**

- Parallelism
- Register pressure
- Code size
- Execution time

❑ **Effectiveness/Cost of the technique**

- ❑ **Value of the code generated**

- Parallelism
- Register pressure
- Code size
- Execution time

- Effectiveness on exploiting ILP
- What is the difference between II and MII?

- ❑ **Effectiveness/Cost of the technique**



❑ **Value of the code generated**

○ Parallelism

○ Register pressure

○ Code size

○ Execution time

- Software pipelining puts high demands on register pressure
- How many regs are needed?
- How many loops within a given number of registers?

❑ **Effectiveness/Cost of the technique**

- ❑ **Value of the code generated**

- Parallelism
- Register pressure
- **Code size**
- Execution time

- Crucial in embedded domains
- Stages of a schedule

- ❑ **Effectiveness/Cost of the technique**



❑ **Value of the code generated**

- Parallelism
- Register pressure
- Code size
- Execution time

• **Main objective**

❑ **Effectiveness/Cost of the technique**



❑ **Value of the code generated**

- Parallelism
- Register pressure
- Code size
- Execution time

❑ **Effectiveness/Cost of the technique**

- Can all the loops be scheduled?
- Compilation time



# Talk Outline

- Modulo Scheduling Background
- Selection Criteria
- **Techniques Compared**
- Study Environment
- Results
- Conclusions

## Stage MS

- Post-pass that can be applied after a MS technique
- To reduce the Register Pressure
- Without increasing the II
- Moves operations by II
- Various heuristics. We selected 3UP+RSS heuristic



# Main Differences

## TECHNIQUES COMPARED

	IMS	SMS	Slack MS	IRIS
Order of nodes	Top-Down	<ul style="list-style-type: none"><li>• Priority to recurrences</li><li>• No pred. and succ. scheduled in partial schedule</li></ul>	<ul style="list-style-type: none"><li>• Dynamic</li><li>• Based on Slack</li></ul>	Top-Down
Finding a cycle	Top-Down	<ul style="list-style-type: none"><li>• Bi-directional</li><li>• Close to pred or succ</li></ul>	<ul style="list-style-type: none"><li>• Bi-directional</li><li>• Close to pred or succ. depending on the benefit</li></ul>	Stage MS Heuristics
Backtracking	Yes	No	Yes	Yes



# Qualitative Comparison

TECHNIQUES COMPARED

	IMS	SMS	Slack MS	IRIS
Parallelism	Backtracking	Order	<ul style="list-style-type: none"><li>•Order</li><li>•Backtracking</li></ul>	Backtracking
Register Pressure	No	<ul style="list-style-type: none"><li>•Order</li><li>•Bi-directional</li></ul>	Bi-directional	Stage Heuristics
Code Size	Yes	Yes	Yes	Yes
Effectiveness	Backtracking		Backtracking	Backtracking
Cost		No Backtracking		





# Talk Outline

- Modulo Scheduling Background
- Selection Criteria
- Techniques Compared
- **Study Environment**
- Results
- Conclusions

- ❑ **Platform (i.e. compiler)**

ICTINEO

- ❑ **Benchmarks**

- SPECfp95

- Perfect Club

***1936 loops***

- ❑ **Target architectures**

- Some architectures varying the complexity

- ✓ Low Complexity architecture

- ✓ Medium Complexity architecture

- ✓ Complex architecture

**Less Constrained**



**More Constrained**



# Architectures Description

STUDY ENVIRONMENT

Low Complexity	Medium Complexity	Complex Architecture
Fully Pipelined ops		Fully Pipelined Simple ops Non-Pipelined Complex ops
8-Issue	4-Issue	
Unlimited register ports		4 write-ports 8 read-ports
2 memory ports		
2 Int FU and 2 FP FU		

Latencies		Low/Medium	Complex
MEM		2	3
INT	ADD, SUB, COMP	1	3
	MUL	2	4
	DIV, MOD, SQRT	6	8
FP	ADD, SUB, COMP	3	5
	MUL	6	8
	DIV, MOD, SQRT	18	20



# Methodology

- **Study of the BudgetRatio for each architecture: 1, 2.5, 5 and 10**
  - Effectiveness
  - Performance
  - Cost
  
- **Measures for each technique with and without Stage MS**
  - Effectiveness and cost
  - Parallelism
  - Register pressure
  - Code size
  - Execution



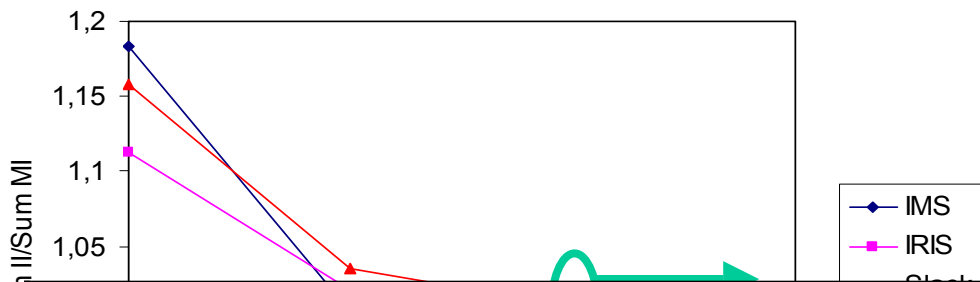
# Talk Outline

- Modulo Scheduling Background
- Selection Criteria
- Techniques Compared
- Study Environment
- **Results**
- Conclusions

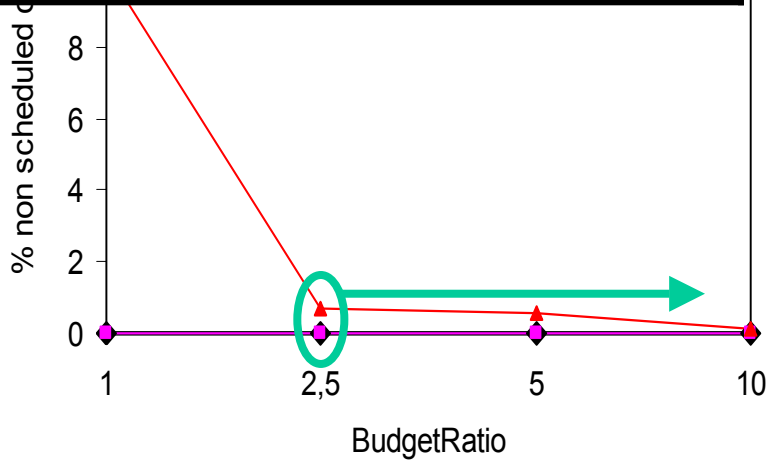
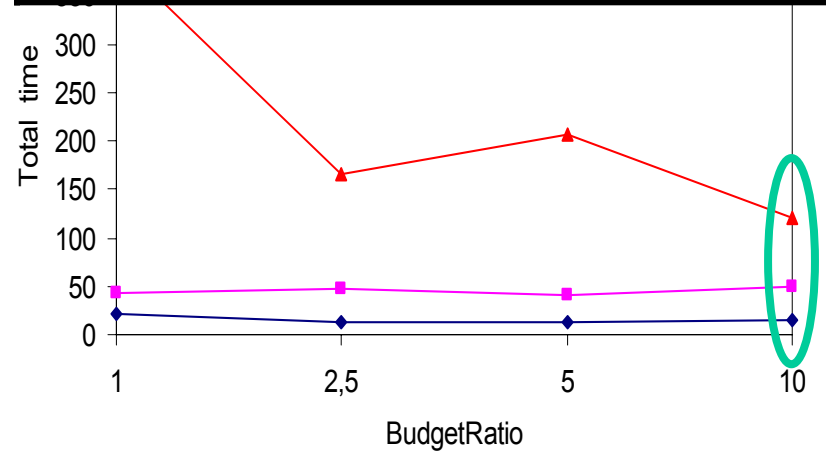


# BudgetRatio Study

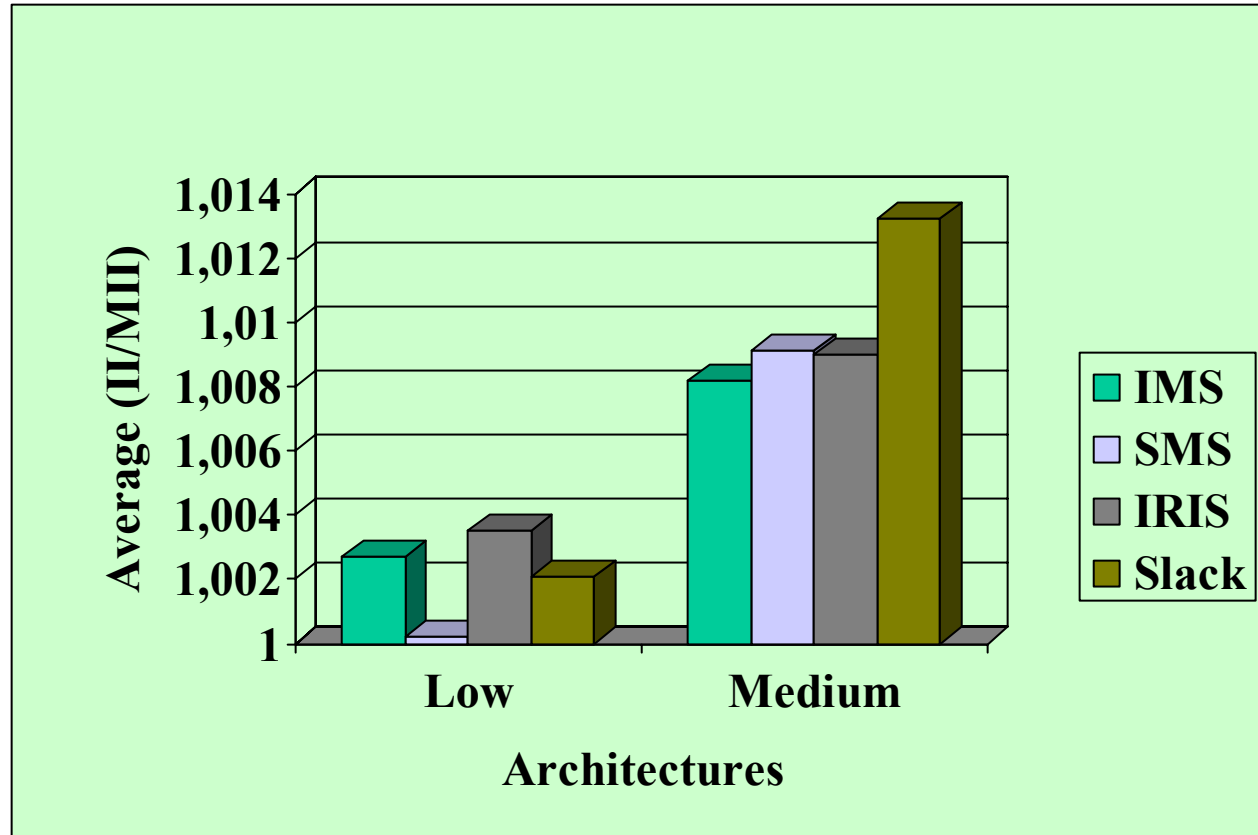
Low Complexity Architecture



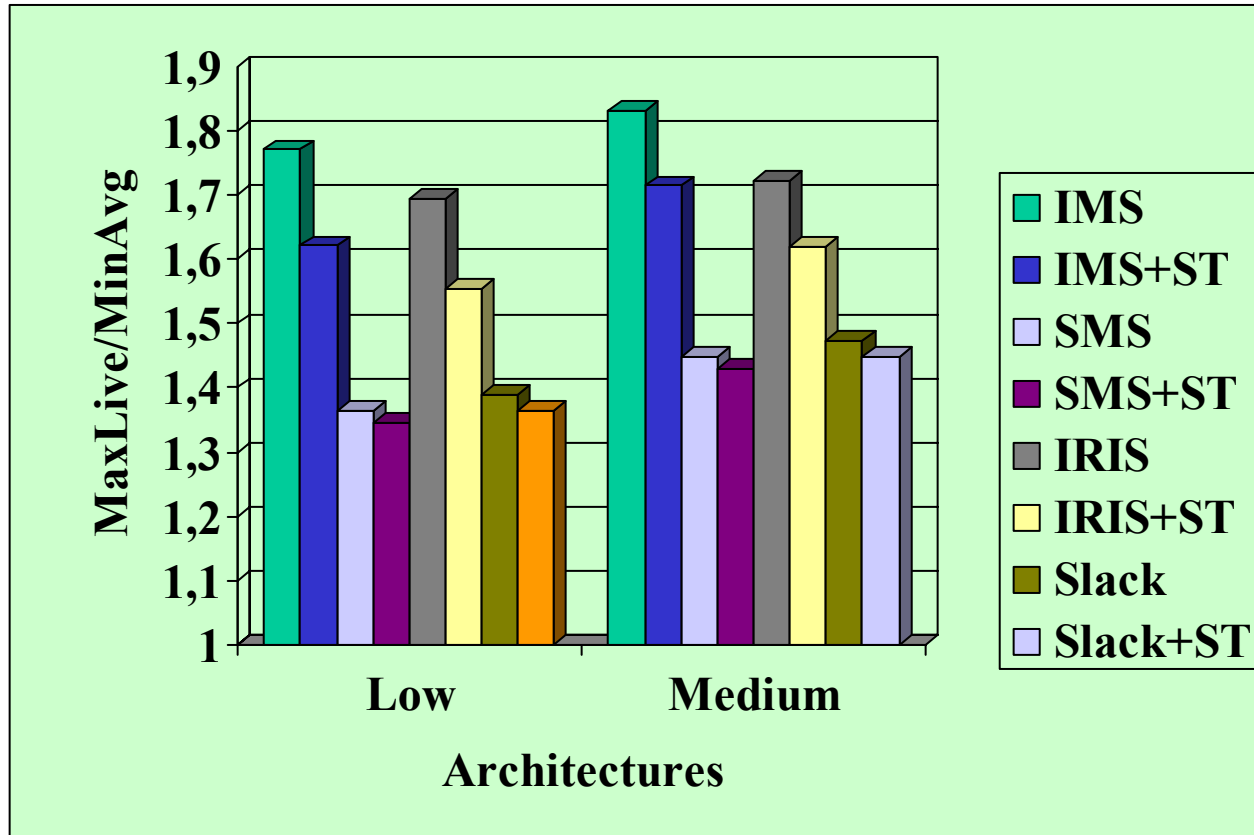
Complexity	BudgetRatio
Low Complexity	10
Medium Complexity	5
Complex Architecture	2.5



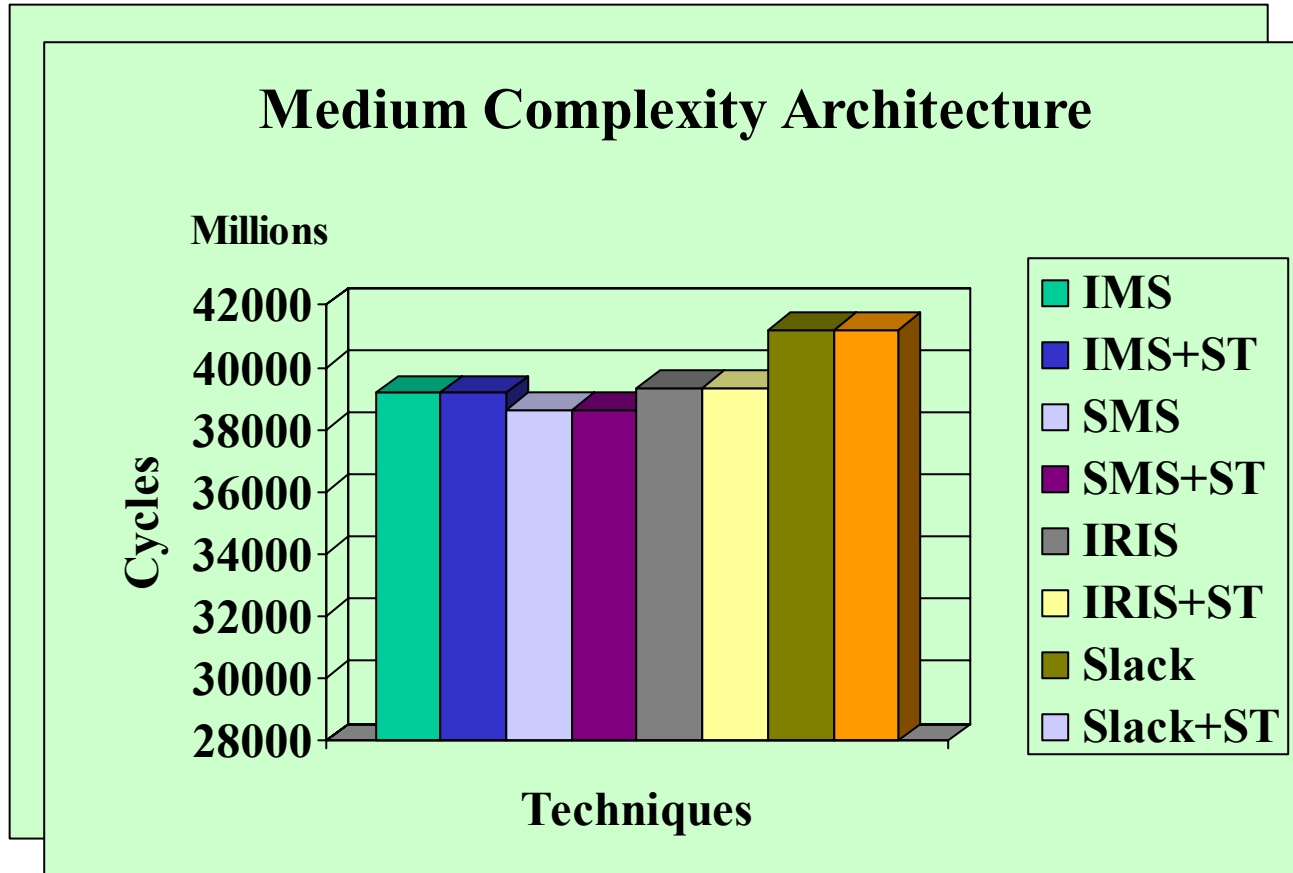
RESULTS

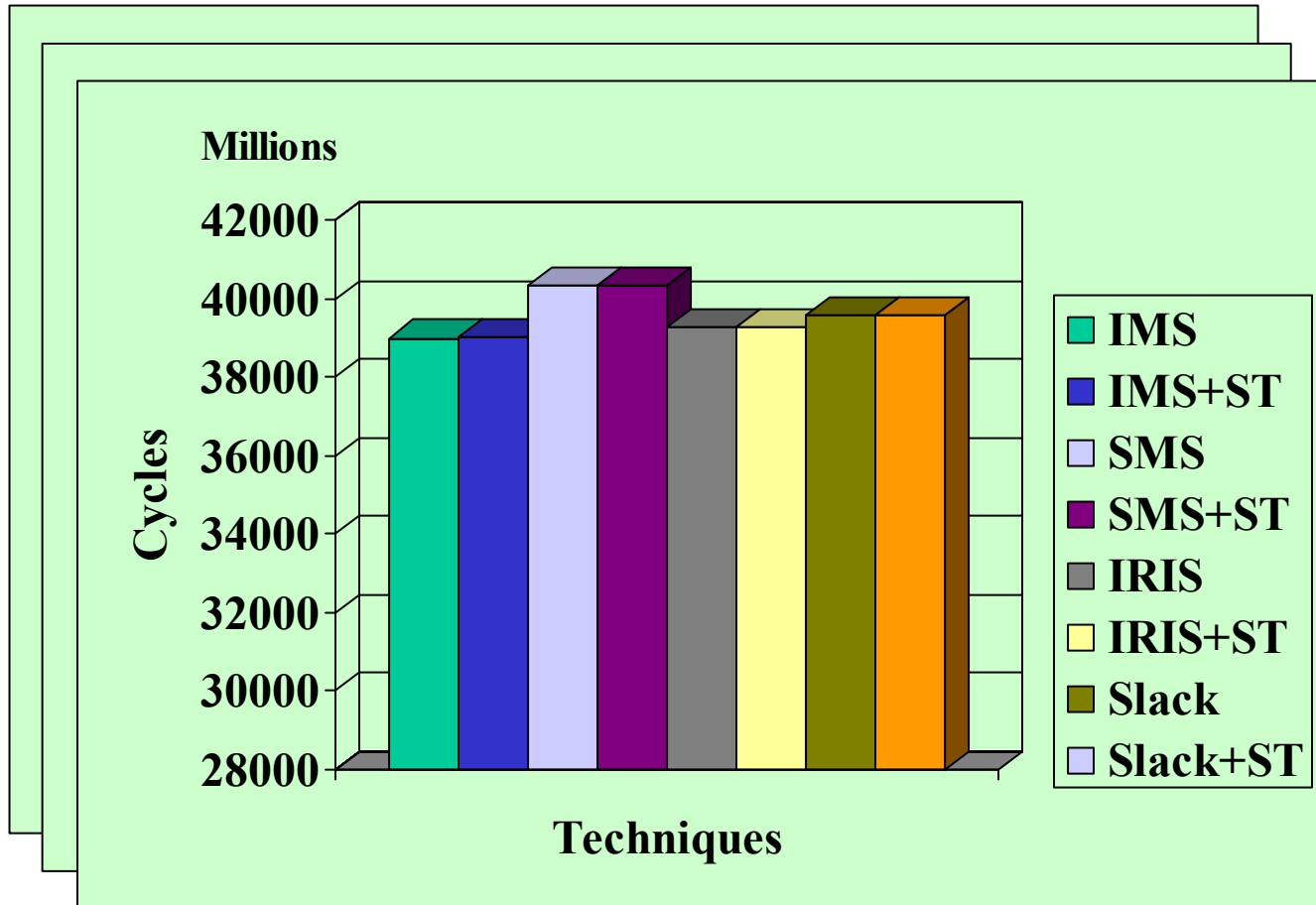


# Register Pressure











# Talk Outline

- Modulo Scheduling Background
- Selection Criteria
- Techniques Compared
- Study Environment
- Results
- **Conclusions**

- ❑ **We have performed a comparison of well known techniques**
  - IMS, SMS, Slack MS, IRIS and Stage MS
  
- ❑ **We have use a common**
  - Compiler platform
  - Benchmarks
  - Target architectures
  - Measures
  
- ❑ **To perform a Quantitative Comparison**
  - Study of the BudgetRatio
  - Measures of the techniques (Code quality, Effectiveness and cost)



# Results Conclusion

## ■ **Some interesting results:**

- SMS best II and register pressure for lower/medium complex
- IMS slightly better II for complex architectures
- Slack MS, close to SMS in register pressure
- IMS and IRIS improved by Stage MS but still beyond SMS and Slack

## ■ **Bottomline**

- SMS is generally the best or close to the best

## ■ **The tool can be used to assess any particular microarchitecture**



UNIVERSITAT POLITÈCNICA  
DE CATALUNYA

# A Comparative Study of Modulo Scheduling Techniques

**Josep M. Codina, Josep Llosa and Antonio González**

Dept. of Computer Architecture  
Universitat Politècnica de Catalunya  
Barcelona, SPAIN

**E-mail: {jmcodina, josepl1, antonio}@ac.upc.es**