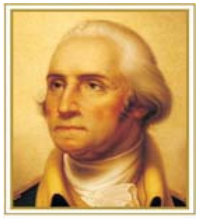


THE GEORGE WASHINGTON UNIVERSITY
WASHINGTON DC

HPCL The George Washington University WASHINGTON DC
High Performance Computing Lab

System-Level Parallelism and Throughput Optimization in Designing Reconfigurable Computing Applications

Esam El-Araby[1], Mohamed Taher[1], Kris Gaj[2], Tarek El-Ghazawi[1], David Caliga[3], and Nikitas Alexandridis[1]

[1]The George Washington University, [2]George Mason University, [3]SRC Computers

Outline

- Background
  - Motivations
  - Problem Statement
- Modeling our Approach
- Experimental Verification
  - Test Bed
  - Experimental Results
- Conclusions

Background

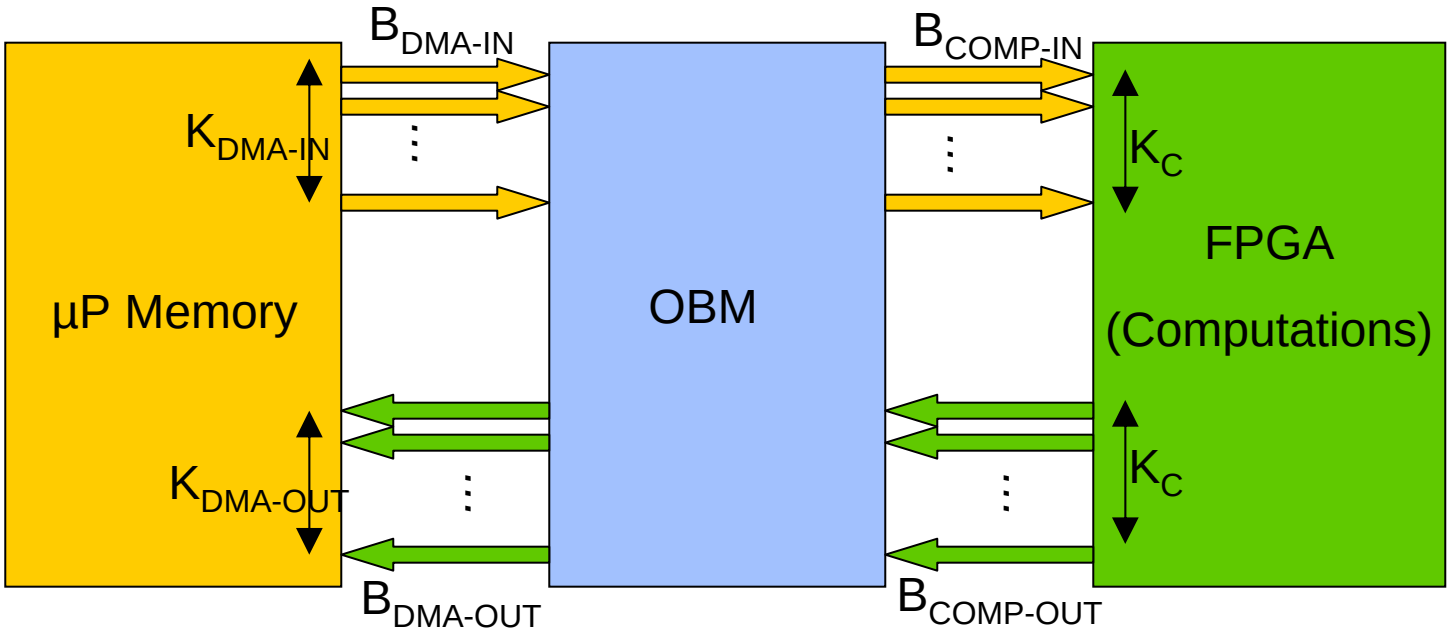
- **Motivations**
  - Considering the reconfigurable computers with multiple microprocessors, FPGA chips, and I/O controllers
  - I/O is the major bottleneck
  - Some remedy can be achieved by overlapping I/O with computations

- **Problem Statement**
  - Given the possibility of running concurrent I/O activities and overlapping them with computations, and
  - Given an application with certain mix of computations and I/O demands, then
  - What would be the best granularity of overlapping I/O with computations, and
  - What would be the expected performance improvement, and
  - Can the results be verified experimentally?

Modeling our Approach

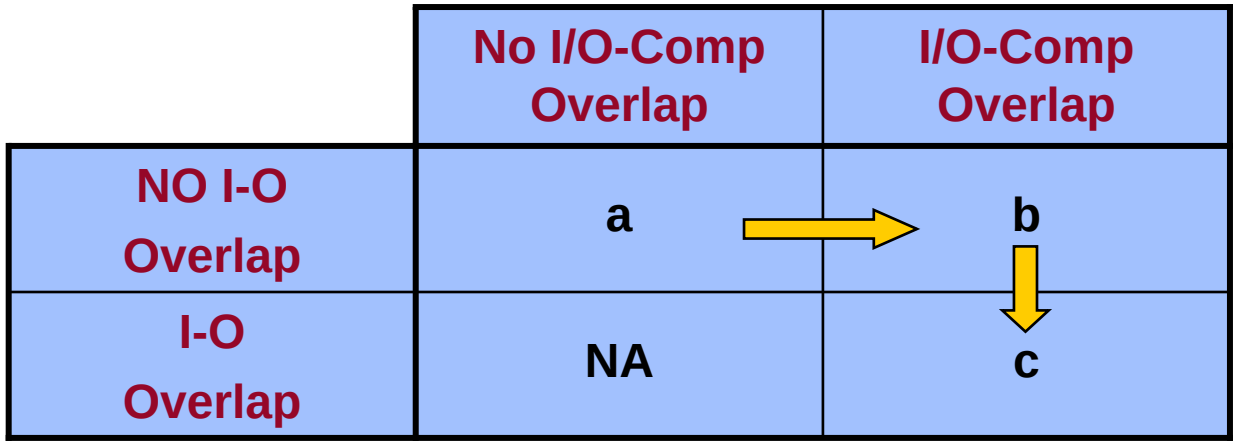
- A model to reflect the implementation performance as a function of platform parameters and application parameters
- Machine Parameters
  - DMA transfers bandwidth
    - Input bandwidth (B_{DMA-IN})
    - Output bandwidth ($B_{DMA-OUT}$)
  - Multiplicity of DMA channels
    - Input multiplicity (K_{DMA-IN})
    - Output multiplicity ($K_{DMA-OUT}$)
    - Channel-Overlapping factor (V)
- Application parameters
  - Computations bandwidth
    - Input bandwidth ($B_{COMP-IN}$)
    - Output bandwidth ($B_{COMP-OUT}$)
  - Computations multiplicity, concurrency, (K_c)
  - Computations to I/O ratio (X_c)
  - Partitionability of data and processing (n)

Model Architecture



- **Platform**
  - **Multi-Channel I/O Transfers**
  - **Asymmetric transfers**
  - **Buffering mechanism (OBM)**
- **Application**
  - **Concurrency**
  - **Data-producing and/or Data-consuming**

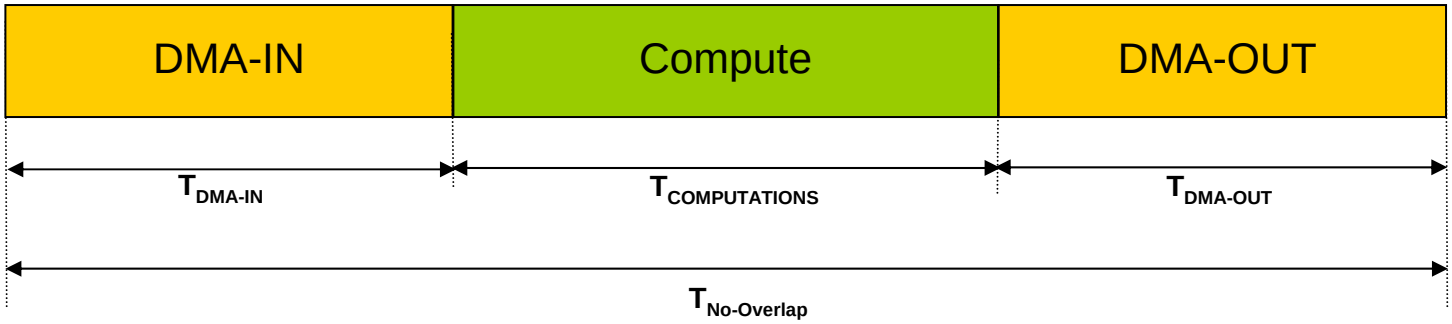
Overlapping Computations with Transfers



- **Possible overlapping scenarios**
  - **A → No overlapping case**
  - **B → Non-overlapped DMA channels**
  - **C → Fully overlapped DMA channels**

No Overlap Scenario

	No I/O-Comp Overlap	I/O-Comp Overlap
NO I-O Overlap	a	b
I-O Overlap	NA	c



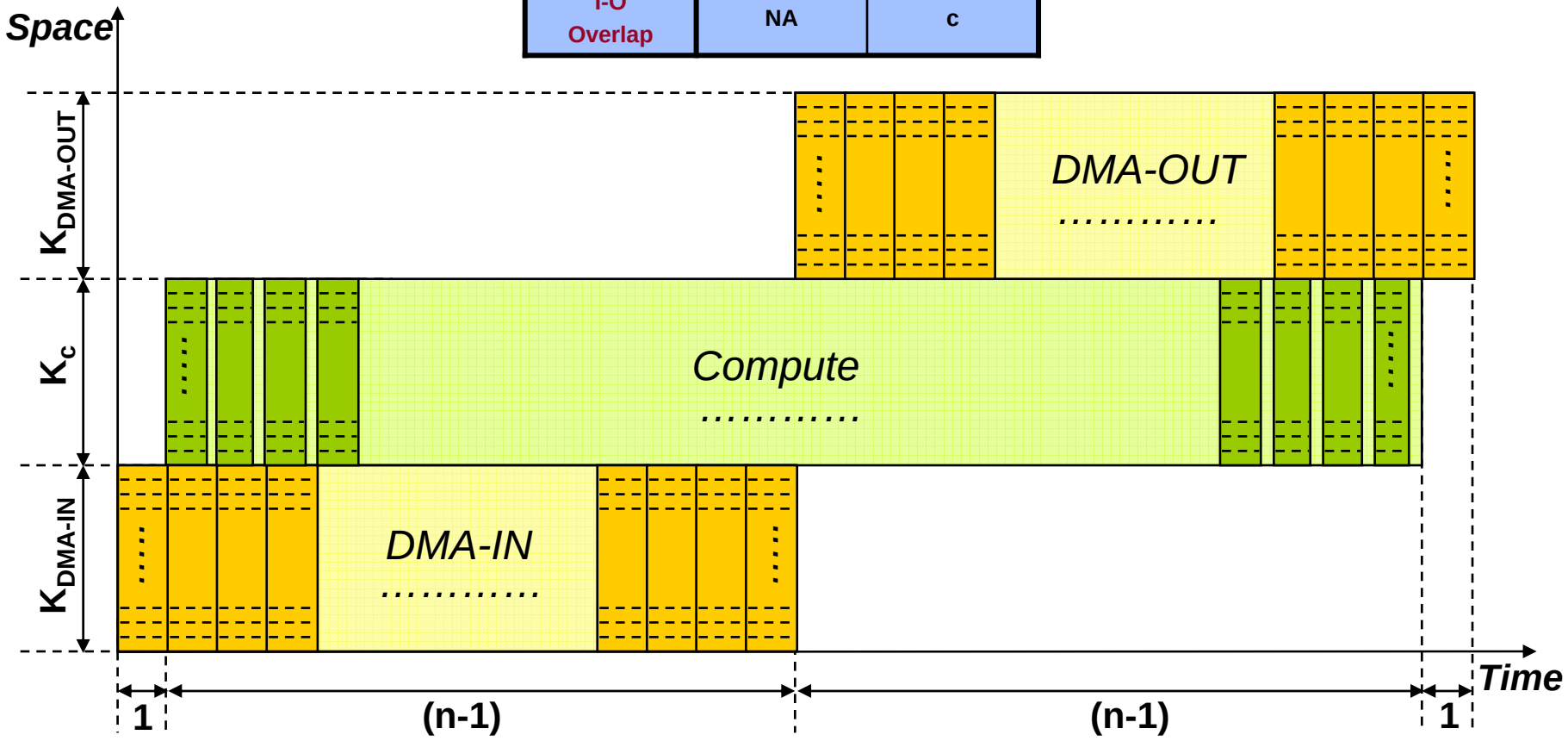
$$T_{DMA} = T_{DMA\text{-}IN} + T_{DMA\text{-}OUT}$$

$$X_{in} = T_{DMA\text{-}IN} / T_{DMA} \quad , \quad X_{out} = T_{DMA\text{-}OUT} / T_{DMA}$$

$$X_c = T_{COMPUTATIONS} / T_{DMA}$$

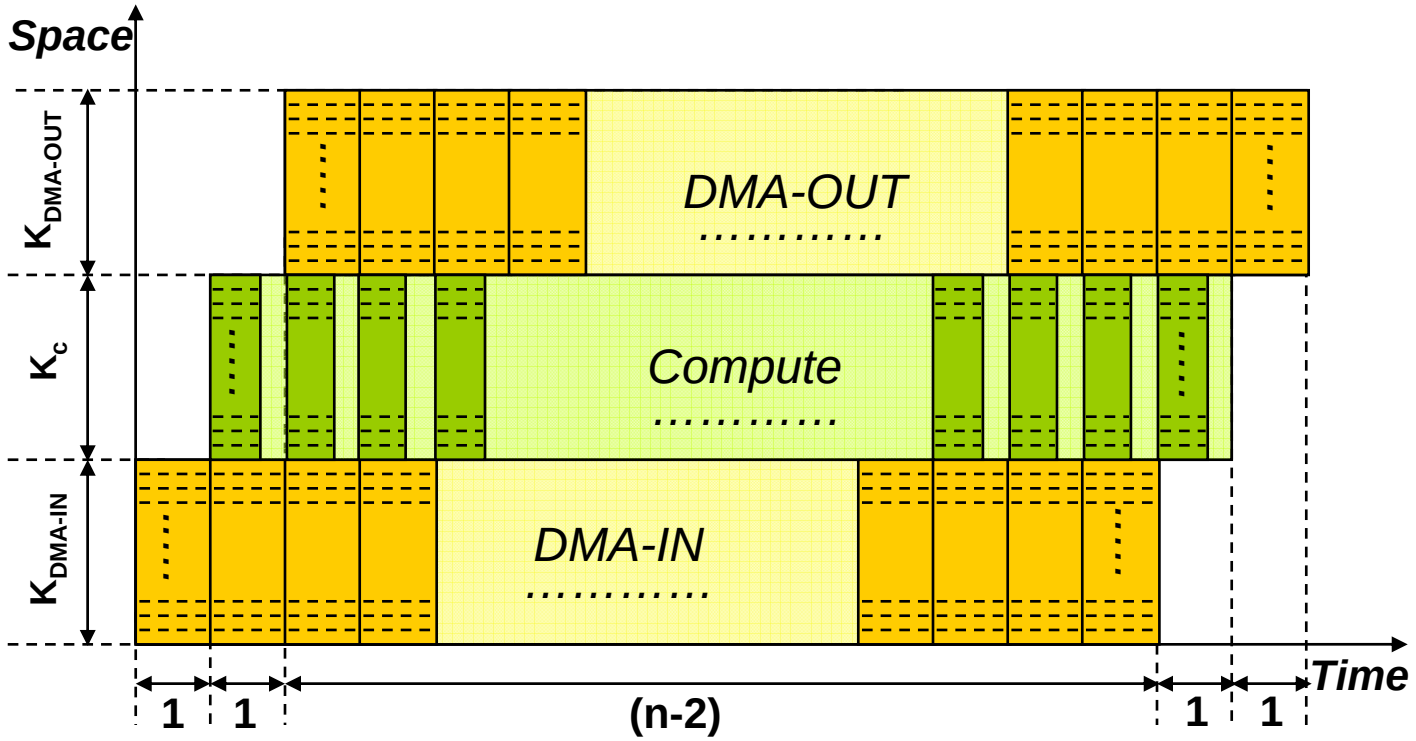
Overlap Scenario
(Non-overlapped Multi-channel DMA, *V*=0)

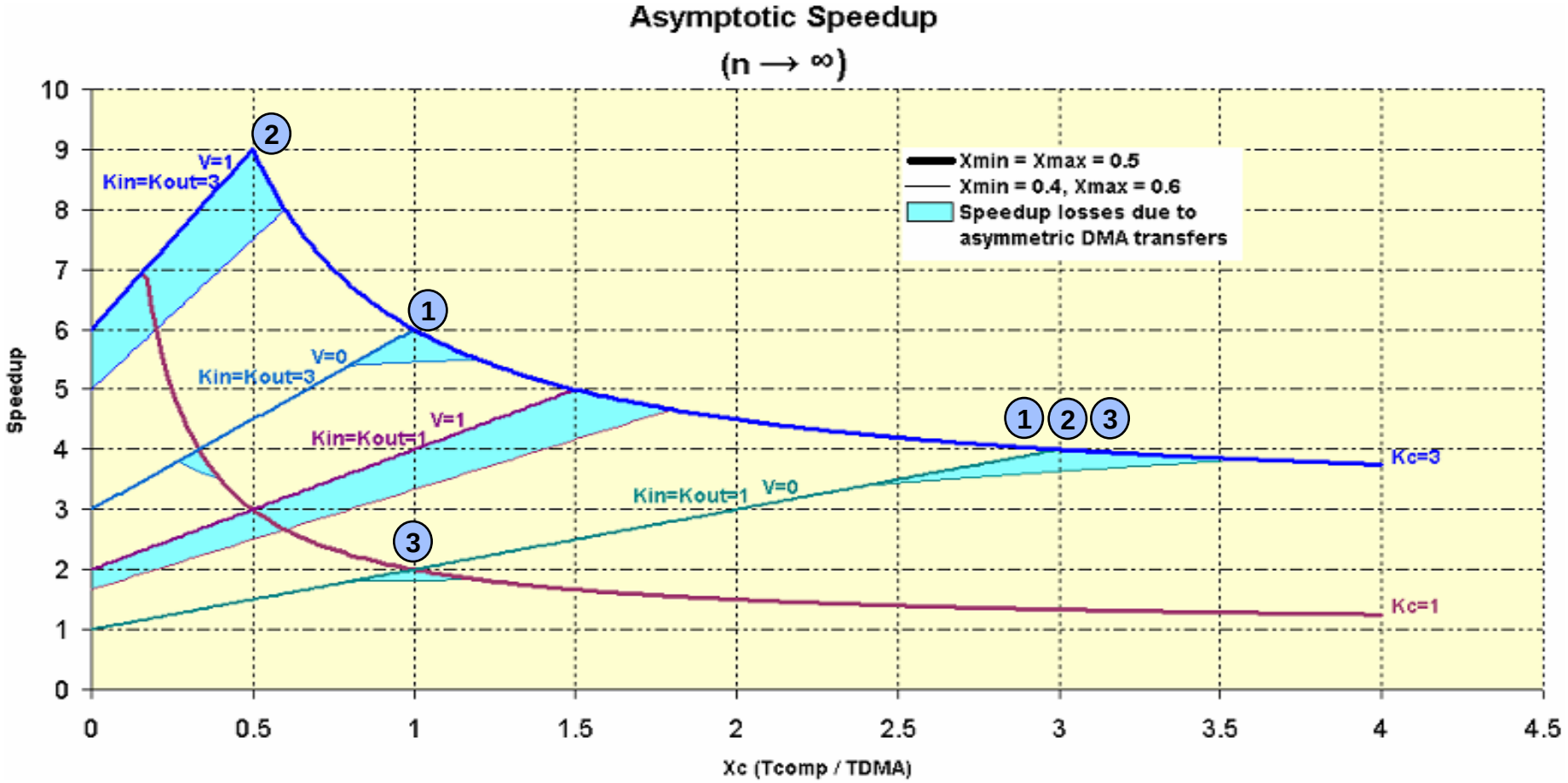
	No I/O-Comp Overlap	I/O-Comp Overlap
NO I-O Overlap	a	b
I-O Overlap	NA	c



Overlap Scenario
(Overlapped Multi-channel DMA, *V*=1)

	No I/O-Comp Overlap	I/O-Comp Overlap
NO I-O Overlap	a	b
I-O Overlap	NA	c



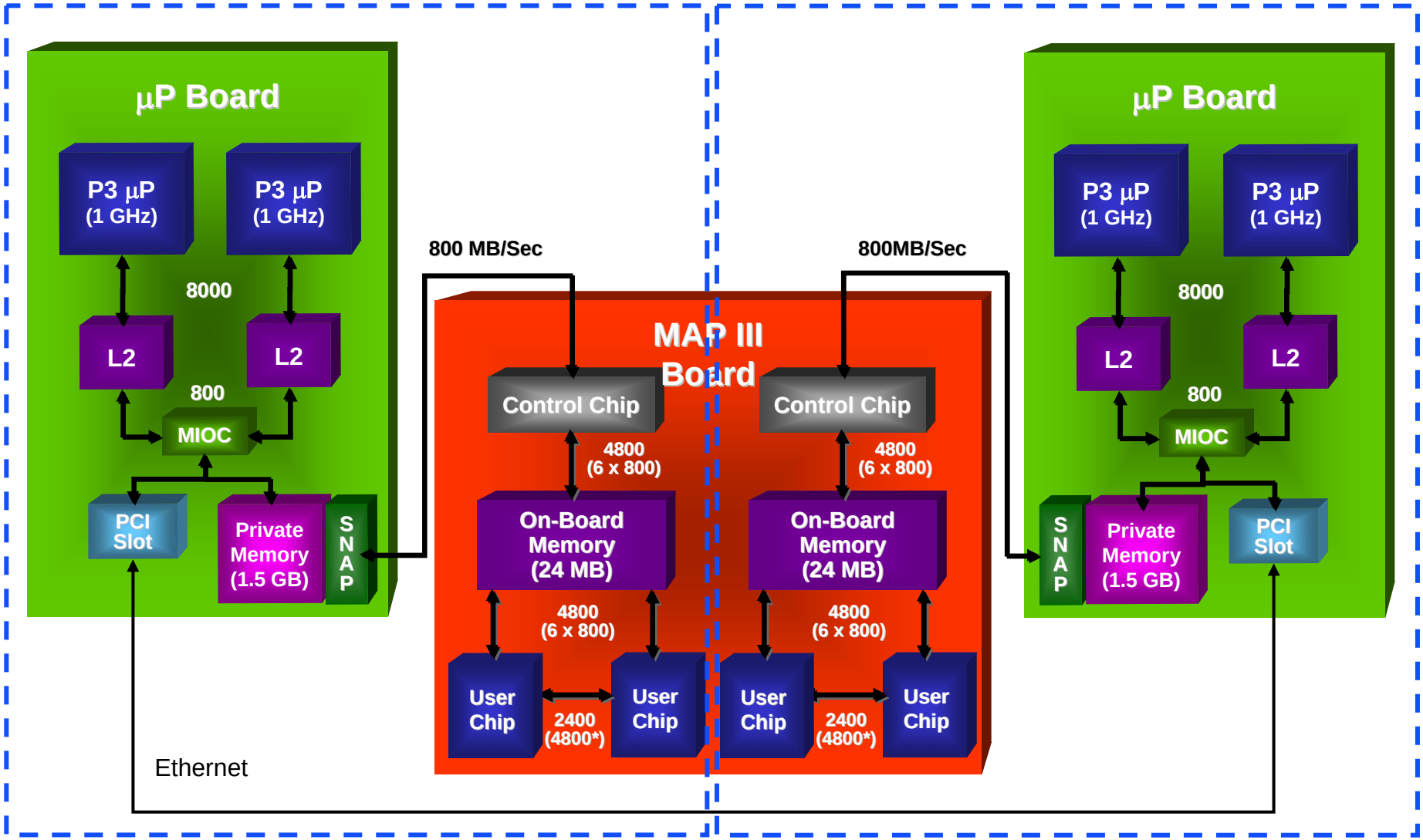


- 1 - Effect of I/O multiplicity ($K_{DM\ A}$)
- 2 - Effect of DMA overlapping factor (V)
- 3 - Effect of computations multiplicity (K_c)

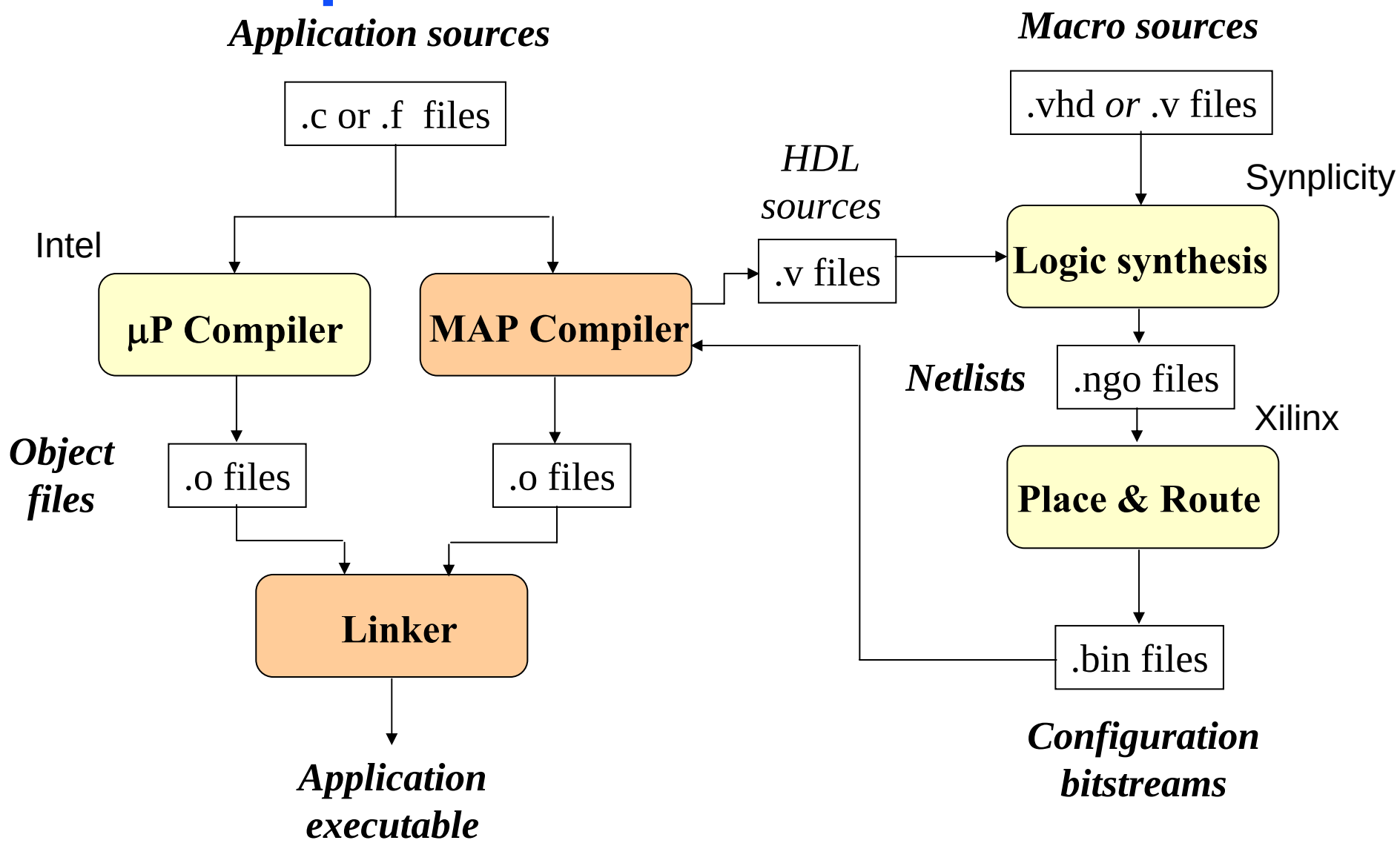
$$S_{\infty max} = \lim_{n \to \infty} S \Big|_{X_c = X_{cmax}} = \begin{cases} K_c + \frac{1}{2} K_{DMA} \ldots\ldots\ldots V = 0 \\ K_c + K_{DMA} \ldots\ldots\ldots V = 1 \end{cases} \qquad X_{cmax} = X_{cmin} = \begin{cases} \frac{2K_c}{K_{DMA}} \ldots\ldots\ldots V = 0 \\ \frac{K_c}{K_{DMA}} \ldots\ldots\ldots V = 1 \end{cases}$$

Experimental Verification

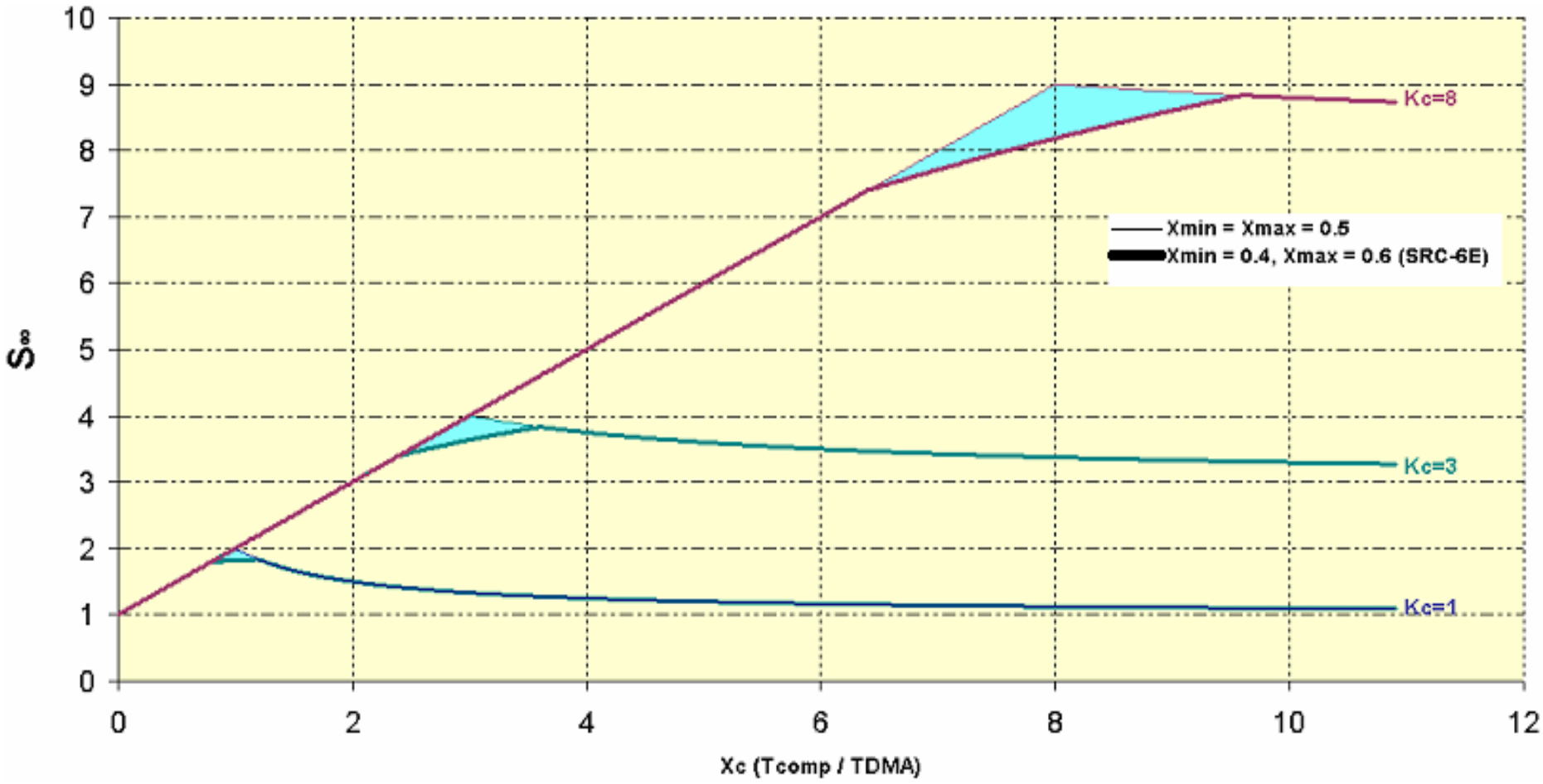
SRC-6E Hardware Architecture



Compilation Process of SRC-6E



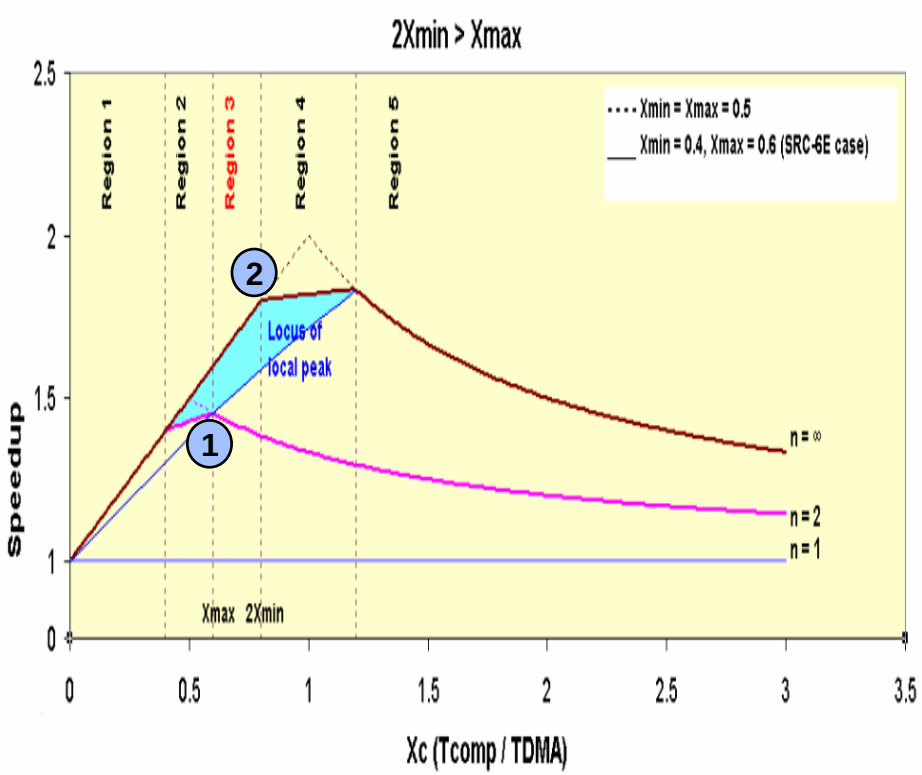
Theoretical Asymptotic Speedup for SRC-6E ($K_{DMA\text{-}IN}=K_{DMA\text{-}OUT}=1$, V=0)

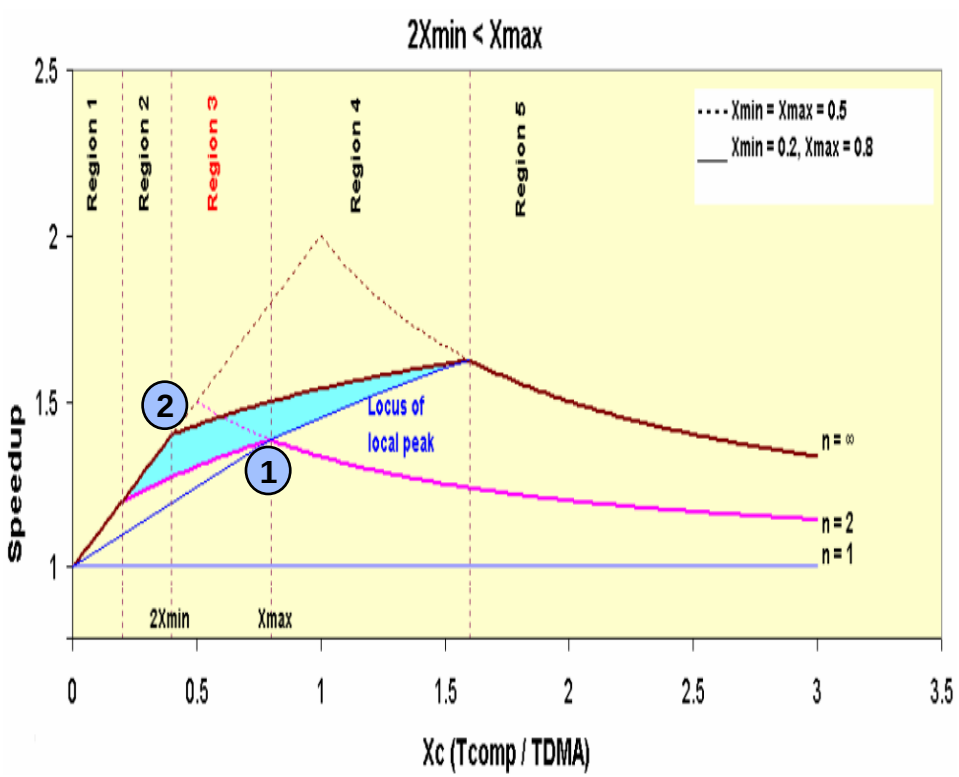


$$S_{\infty \max} = K_c + \frac{1}{2X_{\max}} = K_c + \frac{1}{2X_{out}} = K_c + 0.83$$

The Design Problem

- **The design problem can be stated as follows:**
  - **Given the machine constraints and the application constraints, what is the minimum number of transfer parcels that achieves a speedup as close as possible to the asymptotic maximum for that application?**
- **In other words, given X_{in}, X_{out}, $K_{DMA\text{-}IN}$, $K_{DMA\text{-}OUT}$, V, and K_C, we are trying to find the minimum n, n_{min}, that gives speedup S very close to S_∞ with an efficiency E near to 1, where S_∞ is the asymptotic value of S for this specific application, and E is the ratio between S and S_∞**



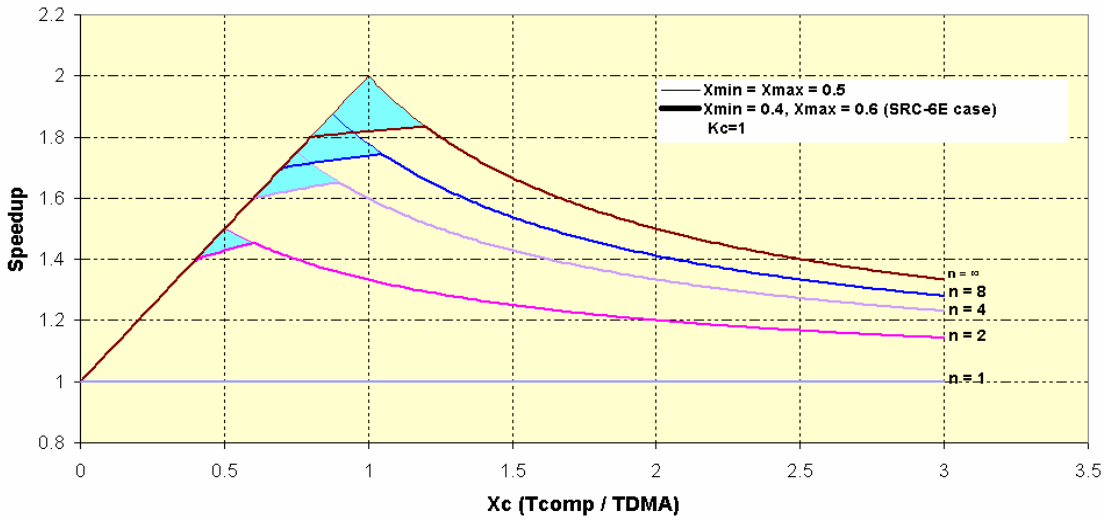


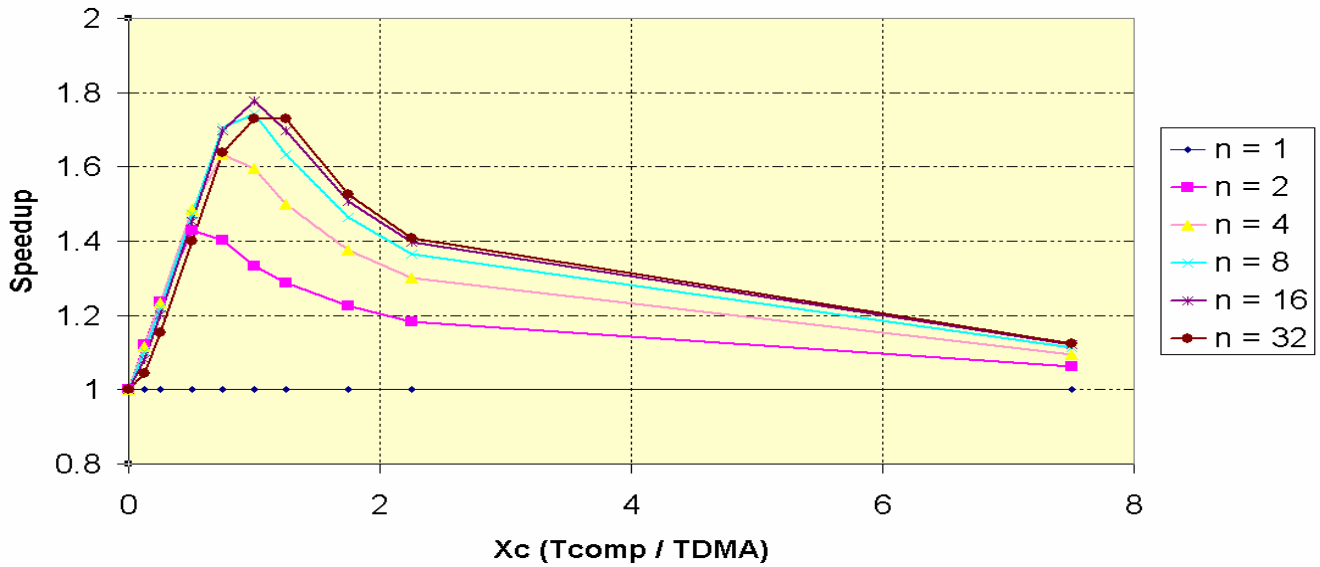
Design values for the minimum number of transfer parcels, n_{min}

	Region 1	Region 2		Region 3		Region 4		Region 5
		$2X_{min} > X_{max}$	$2X_{min} < X_{max}$	$2X_{min} > X_{max}$	$2X_{min} < X_{max}$	$2X_{min} > X_{max}$	$2X_{min} < X_{max}$	
	$0 \le X_c \le X_{min}$	$X_{min} \le X_c \le X_{max}$	$X_{min} \le X_c \le 2X_{min}$	$X_{max} \le X_c \le 2X_{min}$	$2X_{min} \le X_c \le X_{max}$	$2X_{min} \le X_c \le 2X_{max}$	$X_{max} \le X_c \le 2X_{max}$	$2X_{max} \le X_c < \infty$
S_∞	$1+X_c$	$1+X_c$		$1+X_c$	$\dfrac{1+X_c}{X_{max}+\frac{1}{2}X_c}$	$\dfrac{1+X_c}{X_{max}+\frac{1}{2}X_c}$		$1+\dfrac{1}{X_c}$
S_p	S_∞	$\dfrac{2(1+X_c)}{1+X_{max}+X_c}$		$\dfrac{1+X_c}{1+(1-\frac{1}{2X_{max}})X_c}$	$\dfrac{2(1+X_c)}{1+X_{max}+X_c}$	$\dfrac{1+X_c}{1+(1-\frac{1}{2X_{max}})X_c}$		$\dfrac{1+X_c}{\frac{1}{2}+X_c}$
E_p	1	S_p/S_∞		S_p/S_∞	S_p/S_∞	S_p/S_∞		S_p/S_∞
n_{min} E=1	2	$\dfrac{1}{1-\frac{X_c}{2X_{min}}}$		$\dfrac{1}{1-\frac{X_c}{2X_{min}}}$	∞	∞		∞
n_{min} $E=E_p$					$\dfrac{EpX_{min}}{(1-Ep)(X_{max}+\frac{1}{2}X_c)}$	$\dfrac{EpX_{min}}{(1-Ep)(X_{max}+\frac{1}{2}X_c)}$		$\dfrac{Ep}{(1-Ep)X_c}$

Experimental Results

Theoretical Speedup





Conclusions

- An overlapping technique is introduced for optimizing the performance of applications on reconfigurable computers
- This overlapping requires dividing data transfers into multiple transfer parcels that can be overlapped with partial computations
- A mathematical model for this technique has been derived for a generic reconfigurable machine, taking into account the constraints imposed by both the system and the application
- The maximum theoretical speedup was shown to be the sum of the I/O channel multiplicity and the application multiplicity, for symmetric I/O transfers
- The model can be used for either designing HW for a fixed application or for designing applications for a fixed HW

Conclusions (cnt'd)

- The presented technique has been implemented and experimentally verified on the SRC-6E reconfigurable computer
- For the current generation of the SRC system, the theoretical maximum speedup was shown to be 1.83, and the corresponding experimental maximum speed-up was 1.78
- The next generation of SRC system has been reported to go beyond this limit