INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)

Dundigal, Hyderabad - 500 043

COMPUTER SCIENCE AND ENGINEERING

TUTORIAL QUESTION BANK

Course Name	High Performance Architecture
Course Code	**BCS003**
Class	I M.Tech
Branch	CSE/IT
Year	2017 - 2018
Team of Instructors	R.M.Noorullah

OBJECTIVES

To meet the challenge of ensuring excellence in engineering education, the issue of quality needs to be addressed, debated and taken forward in a systematic manner. Accreditation is the principal means of quality assurance in higher education. The major emphasis of accreditation process is to measure the outcomes of the program that is being accredited.

In line with this, faculty of Institute of Aeronautical Engineering, Hyderabad has taken a lead in incorporating philosophy of outcome based education in the process of problem solving and career development. So, all students of the institute should understand the depth and approach of course to be taught through this question bank, which will enhance learner's learning process.

COURSE OBJECTIVES:

The course should enable the students to:

I.	Understand the compiling issues for various parallel architectures.
II.	Implementation of transformation technology for code parallelization.
III.	Familiar with the concepts of Data dependence, loop normalization, ZIV,SIV,MIV testing, fine grained for loop distribution, course grained by privatization, handling flow control and improving register reuse.
IV.	Understand the memory management and scheduling for code parallelization.
V.	Understand the optimizing compiler performance for High Performance.

COURSE LEARNING OUTCOMES:

Students, who complete the course, will have demonstrated the ability to do the following:

BCS003.01	Understand the key concerns that are common to improve the performance of compiler.
BCS003.02	Describe Compiling for scalar, super scalar, VLIW, vector and parallel processor.

BCS003.03	Memorizing Bernstein's condition to execute parallel processing.
BCS003.04	Describe the concept of Data Dependence, types and loop carried and loop independent dependence.
BCS003.05	Understand Loop normalization , parallelization , vectorization and scalar renaming.
BCS003.06	Demonstrate Simple dependence testing and subscript portioning.
BCS003.07	Describe the concept of single subscript and multiple induction variable tests.
BCS003.08	Understand the importance of Delta test in testing coupled group.
BCS003.09	Understand the concept more powerful and multiple simple test.
BCS003.10	Describe the overall dependence testing.
BCS003.11	Memorize fine grained and enhancing fine grained by using loop distribution.
BCS003.12	Understand the principles of loop interchange for vectorization.
BCS003.13	Describe the course grained and enhancing by using privatization.
BCS003.14	Understand loop interchange for parallelization.
BCS003.15	Describe how to handle control flow by using if-conversion.
BCS003.16	Describe the concepts of memory hierarchy used in parallelization for improving performance.
BCS003.17	Understand the concepts of scalar register allocation and cache memory management.
BCS003.18	Implement scalar replacement techniques to optimizing compilers.
BCS003.19	Understand the concept of unroll-and-jam.
BCS003.20	Describe cache blocking and perfecting in increasing performance of parallel architecture.
BCS003.21	Understand to improving register usage by scalar register allocation.
BCS003.22	Understand the concept of data dependence for register reuse.
BCS003.23	Describe the scheduling and tracking in Risk Mitigation, Monitoring and Management Plan.
BCS003.24	Understand loop carried and loop dependent reuse in increasing performance of a compiler.
BCS003.25	Describe pruning dependence graph in register reuse to improve performance.

BCS003.26	Demonstrate dependence spanning multiple iterations and loop inter change for register reuse.
BCS003.27	Possess the knowledge and skills for improving performance of compiler and to succeed in national and international level competitive exams.

PART – A (SHORT ANSWER QUESTIONS)

S. No	QUESTIONS	Blooms taxonomy level	Course outcome
	UNIT – I **PARALLEL AND VECTOR ARCHITECTURES**		
1.	**Memorize** the superscalar diagram	Remember	5
2.	**List** Bernstein conditions for detection of parallelism	Remember	5
3.	**Define** scalar data-flow analysis for detection of parallelism	Remember	5
4.	**List** how scalar data-flow analysis can detect and eliminate scalar dependences?	Remember	5
5.	**Identify** the current state of the art in VLIW compilers?	Understand	6
6.	**Describe** (i) superscalar and (ii) super-pipeline concepts.	Understand	5
7.	**Define** Dependability and explain two main measures of it.	Understand	4

S. No	QUESTIONS	Blooms taxonomy level	Course outcome
8.	**State** four important technologies which have led to the improvements in Computer System.	Remember	6
9.	**Define** VLIW.	Remember	6
10.	**List** VLIW architecture.	Remember	6
11.	**Memorize** how VLIW machine restrict the op-codes which may be placed in any 'slot' of its instructions?	Remember	6
12.	**List** direct dependences	Remember	6
13.	**Memorize** different evaluation techniques.	Remember	6
14.	**State** induction substitution.	Apply	4
15.	**Define** scalar remaining?	Understand	4
16.	**List** overhead does the address generation unit of a vector processor remove from the main calculation pipeline?	Understand	4
17.	**Define** vecterization.	Remember	6

PART – B (LONG ANSWER QUESTIONS)

S. No.	QUESTIONS	Blooms Taxonomy Level	Course Outcome
	UNIT-I **PARALLEL AND VECTOR ARCHITECTURES**		
1	**Explain** which overhead does the address generation unit of a vector processor remove from the main calculation pipeline?	Understand	5
2	**Discuss** at least one reason why the architecture of a vector processor improves the performance of programs that operate on vectors and matrices. (Do not re-use the answer to the previous	Understand	5
3	**Identify** which instructions can a 4-way superscalar complete in one cycle?	Understand	5
4	**List** instructions would you *expect* it to complete?	Remember	6
5	**Distinguish** how your answers to the previous two questions different?	Apply	6
6	**State** an example of an instruction sequence in which the performance of a processor might benefit from register renaming	Remember	6
7	**List** would a processor execute both branches of a conditional branch?	Remember	6
8	**Locate** what circumstances will more instructions enter a processor pipeline than are ever completed('graduated')?	Understand	6
9	A 4-bit branch mask companies every instruction in the MIPS R10000 machine's pipeline . **List** its purpose.	Remember	6
10	**List** whether an instruction can be issued by an instruction issue unit in a superscalar machine?	Remember	1
11	**State** why does a VLIW machine restrict the op-codes which may be placed in any 'slot' of its instructions?	Remember	6
12	**Interpret** which piece of software is crucial in order to achieve good performance from a VLIW machine?	Apply	6
13	**Identify** the main difference between the VLIW and the other approaches to improve performance	Understand	6
14	**List** and explain four important technologies, which have led to the improvements in computer system.	Remember	6
15	**Describe** which piece of software is crucial in order to achieve good performance from a VLIW machine?	Understand	6
16	**Identify** which overhead does the address generation unit of a vector processor remove from the main calculation pipeline?	Apply	6
17	**Solve** the equation for ideal speedup for a superscalar super pipelined processor compared to a sequential processor. Assume N instructions, k-stage scalar base pipeline, superscalar degree of m, and super-pipeline degree of n.	Apply	6
18	**Demonstrate** the capabilities of the instruction fetch/despatch unit needed to make an effective superscalar processor.	Apply	6

PART – C (PROBLEM SOLVING AND CRITICAL THINKING QUESTIONS)

S. No	Question DEPENDENCE TESTING	Blooms Taxonomy Level	Course Outcome
	UNIT – I		
1.	**Determine** the *maximum* number of instructions that this processor can start every cycle? A superscalar processor has 6 functional units.	Understand	3
2.	**List** the functions of the instruction issue unit of a superscalar No marks for "issue instructions" (somewhat obvious!)- list the other functions Processor that the IIU performs	Understand	3
4.	**Draw** a diagram showing how the instruction fetch and the widths of the data path (in words - not bits; your diagram primarily determines performance: the instruction issue width(number of instructions issued per cycle)	Understand	3
5.	**Analyze** the idle cycles in a superscalar processor?	Understand	3

PART – A (SHORT ANSWER QUESTIONS)

S. No	Question	Blooms Taxonomy Level	Course Outcome
	UNIT – II **DEPENDENCE TESTING**		
1.	**Classify** goals of dependence testing.	Understand	3
2.	**Explain** the applications of direction vectors.	Understand	3
3.	**Identify** the applications of distance vectors.	Understand	3
4.	**List** out difference between the direction and distance vectors.	Understand	3
5.	**Classify** indices.	Understand	3
6.	**Describe** subscript.	Understand	3
7.	**Discuss** linearity.	Understand	3
8.	**Explain** conservative testing.	Understand	3
9.	**Describe** Diophantine equation.	Understand	3
10.	**Discuss** complexity?	Understand	3
11.	**Explain** Separability.	Understand	3
12.	**Classify** separable indices.	Understand	3
13.	**Describe** coupled subscript.	Understand	3
14.	**Identify** the importance's of coupled subscript groups.	Understand	3
15.	**Explain** partitions of subscript.	Understand	3
16.	**Describe** ZIV test.	Understand	3
17.	**Discuss** SIV test.	Understand	3
18.	**Explain** MIV test?	Understand	3
19.	**Describe** about strong ZIV test?	Understand	3
20.	**Discuss** about strong SIV test?	Understand	3

PART – B (LONG ANSWER QUESTIONS)

S. No	Question	Blooms Taxonomy Level	Course Outcome
	UNIT – II		
1.	**Explain** conservative testing with an example.	Understand	3
2.	**Define** complexity? Explain in detail with an example.	Apply	4
3.	**Explain** about subscript partition algorithm.	Understand	3
4.	**Execute** ZIV test with example.	Apply	4
5.	**Solve** SIV test with example.	Apply	4
6.	**State** MIV test? Explain with example.	Remember	5
7.	**Demonstrate** conservative testing with an example.	Apply	4
8.	**Explain** why does a VLIW machine need a good optimizing compiler?	Understand	6
9.	**Explain** Where can you find a small dataflow machine in every high performance processor?	Apply	4
10.	**Demonstrate** why does branch prediction speed up a processor?	Apply	4
11.	**Define** the status bits in a branch target buffer.	Remember	5
12.	**Interpret** one Strong SIV Test Example	Apply	4
13.	**Explain** one weak SIV Test Example	Understand	3
14.	**Describe** weak-zero SIV test?	Understand	3
15.	**Implement** Weak-Zero SIV & Loop Peeling	Apply	6
16.	**Execute** Weak-Crossing SIV Test	Apply	6
17.	**Define** Weak-crossing SIV & Loop Splitting	Remember	5

PART – C (PROBLEM SOLVING AND CRITICAL THINKING QUESTIONS)

S. No	Question DEPENDENCE TESTING	Blooms Taxonomy Level	Course Outcome
	UNIT – II		
1.	**How** to Determine whether dependences exist between two subscripted references to the same array in a loop nest, explain in detail with a problem?	Understand	3
2.	**Describe** how to prove that no dependence exists between given pairs of subscripted references to the same array variable. This is the desired outcome.	Apply	4
3.	**Report** the Subscript Partitioning Algorithm.	Understand	3
4.	**Implement** to merge direction vectors with an example	Apply	4
5.	**Interpret** breaking conditions in dependence test with an example?	Apply	4
6.	**State** delta test with an example.	Remember	3
7.	**How** to Determine whether dependences exist between two subscripted references to the same array in a loop nest, explain in detail with a problem?	Apply	4

PART – A (SHORT ANSWER QUESTIONS)

S. No	QUESTIONS	Blooms taxonomy level	Course outcome
	UNIT – III **FINE-GRAINED AND COARSE-GRAINED PARALLELISM**		
1.	**Define** Fine-Grained parallelism.	Remember	5
2.	**Give** some examples of Fine-Grained parallelism.	Remember	5
3.	**Give** some examples of Fine-Grained parallelism.	Remember	5
5.	**Explain** loop skewing?	Remember	5

6.	**List** out uses of loop skewing?	Remember	5
7.	**Define** scalar renaming?	Remember	5
8.	**Describe** array renaming?	Remember	5
9.	**List** out uses of Fine-Grained parallelism.	Remember	5
10.	**Describe** loop distribution.	Remember	4
11.	**Define** Coarse-Grained parallelism.	Remember	5
12.	**Give** some examples of Coarse-Grained parallelism.	Understand	5
13.	Define load imbalance.	Understand	5
14.	**List** out Key features of independent parallelism?	Understand	5

PART – B (LONG ANSWER QUESTIONS)

S. No	QUESTIONS	Blooms taxonomy level	Course outcome
	UNIT – III **FINE-GRAINED AND COARSE-GRAINED PARALLELISM**		
1.	**Explain** in detail about Fine-Grained parallelism.	Understand	3
2.	**List** out uses of Fine-Grained parallelism.	Understand	3
3.	**Differentiate** between Fine-Grained and Coarse-Grained		3
2.	**Describe** loop distribution.	Understand	3
3.	**Define** Coarse-Grained parallelism.	Understand	3
4.	**Give** some examples of Coarse-Grained parallelism.	Understand	3
5.	**Describe in** detail about load imbalance technique.	Remember	5
6.	**Explain** in detail about loop skewing.	Remember	5
7.	**Describe** about differences between coarse-grained and fine-grained with diagram?	Remember	5

8.	**Explain** in detail about Coarse and very coarse-grained parallelism?	Remember	5
9.	**Describe** how to Enhance Coarse-Grained parallelism?	Remember	5
10.	**Explain** in detail about privatization which used to enhance Coarse-Grained parallelism	Remember	5
11.	**Explain** in detail about scalar expansion which used to enhance Coarse-Grained parallelism.	Remember	5
12.	Explain about loop alignment technique?	Understand	3
13.	**Describe** about loop fusion technique?	Understand	3

PART – C (PROBLEM SOLVING AND CRITICAL THINKING QUESTIONS)

S. No	QUESTIONS	Blooms taxonomy level	Course outcome
	UNIT – III **FINE-GRAINED AND COARSE-GRAINED PARALLELISM**		
1.	**Explain** in detail about the best approach to use multiple services inside a resource controller?	Understand	3
2.	**How** do you determine how coarse or fine-grained a responsibility should be when using the single responsibility principle?	Apply	4
3.	**Discuss** on Types of synchronization granularity?	Apply	4
2.	**Explain** in detail about how to Enhance Coarse-Grained parallelism with an example?	Understand	3
3.	**Explain** in detail impact of granularity on performance?	Apply	4
4.	**Describe** how to enhance Coarse-Grained parallelism using privatization?	Understand	2
5.	**Describe** how to enhance Coarse-Grained parallelism using scalar expansion?	Apply	3

PART – A (SHORT ANSWER QUESTIONS)

S. No	QUESTIONS	Blooms taxonomy level	Course outcome
	UNIT – IV		
	HANDLING CONTROL FLOW		
1.	**Define** If-conversion.	Understand	3
2.	**Define** scalar-register allocation.	Understand	3
3.	**Explain** cache memory hierarchy.	Understand	3
2.	**Define** scalar replacement.	Understand	3
3.	**Describe** about unroll-and-jam	Understand	3
4.	**Define** loop alignment.	Understand	3
5.	**Define** cache blocking.	Remember	3
6	**Explaining** about perfecting.	Understand	4
7	**Define** bad cache alignment.	Remember	3
8.	**Define** unroll-and-jam in memory hierarchy?	Understand	4

PART – B (LONG ANSWER QUESTIONS)

S. No	QUESTIONS	Blooms taxonomy level	Course outcome
	UNIT – IV		
	HANDLING CONTROL FLOW		
1.	**Draw** the diagram of memory hierarchy.	Understand	5
2.	**Draw** the diagram of cache lines.	Understand	5
3.	**Explain** in detail about cache lines with an example?		5
2.	**Describe** cache blocking with neat diagram?	Understand	6
3.	**Discuss** on loop alignment with an example?	Understand	5
4.	**Describe** how unblocked loop reduces 120 misses. Explain with	Understand	4
5.	**Explain** Cache Use in Stencil Computations with an example?	Remember	5
6.	**List** out the steps involved in using a machine learning technique for building heuristics for program transformation?	Understand	4

PART – C (PROBLEM SOLVING AND CRITICAL THINKING QUESTIONS)

S. No	QUESTIONS	Blooms taxonomy level	Course outcome
	UNIT – IV		
HANDLING CONTROL FLOW			
1.	**Explain** in detail about Cache Use in Stencil Computations	Understand	5
2.	**Explain** how the data move from CPU to hard disc with diagram	Understand	5
3.	**Explain** how the data move from register to another register with diagram		5
4.	**Describe** how One-dimensional blocking reduced misses from 120 to 80? Explain with an example?	Understand	6
7.	**Explain** is it possible to learn a decision rule that selects the parameters involved in loop unrolling efficiency?	Remember	5
8.	**Explain** why does the machine learning Based heuristics achieve better performance than existing ones?	Understand	4
9.	**Describe** why does learning process really take into account the target architecture?		

PART – A (SHORT ANSWER QUESTIONS)

S. No	QUESTIONS	Blooms taxonomy level	Course outcome
	UNIT – V		
IMPROVING REGISTER USAGE			
1.	Define Loop Interchange for Register Reuse.	Understand	5
2.	Define True dependence?	Understand	5
3.	Define Output dependence?		5
4.	Explain Antidependence?	Understand	6
5.	Define Forward carried?	Understand	5
6.	Differentiate between Loop Carried and Loop independent dependence?	Understand	4
7.	Explain why Loop Fusion is Profitable for Register Reuse.	Remember	5
8.	Explain two cases where Loop Fusion is profitable.	Understand	4

9.	Explain forward loop carried dependence.	Understand	4
10.	Define what is Backward carried.	Understand	4

PART – B (LONG ANSWER QUESTIONS)

S. No	QUESTIONS	Blooms taxonomy level	Course outcome
	UNIT – V		
	IMPROVING REGISTER USAGE		
1.		Understand	5
2.	Describe in detail about Loop Fusion for Register Reuse.	Understand	5
3.	Explain with an example on Loop Interchange for Register Reuse		5
4.	Explain in detail scalar replacement with an example.	Understand	6
5.	Write down Loop Interchange Algorithm with and explain in detail with an example.	Understand	5
6.	Apply Loop fusion on following code and transform into better code: A(1:N) = C(1:N) + D(1:N) B(1:N) = C(1:N) – D(1:N)	Understand	4
7.	Explain in detail with an example code why scalar replacement saves the fetching time.	Remember	5
8.	Give example code for forward loop carried dependence	Understand	4
9.	Explain why We cannot simply fuse the two loops with an example code.		
10.	How to fuse two loops without any problem explain in detail.		

PART – C (PROBLEM SOLVING AND CRITICAL THINKING QUESTIONS)

S. No	QUESTIONS	Blooms taxonomy level	Course outcome
	UNIT – V		
	IMPROVING REGISTER USAGE		
1.	Explain in detail why more dependences are merry?	Apply	5
2.	Apply Unroll and jam technique on following code to convert into Unroll and jam? Original Code DO I = 1, N*2 DO J = 1, M A(I) = A(I) + B(J) ENDDO ENDDO	Apply	5

3.	Apply scalar replacement technique on following code and find out corresponding code? DO I = 1, N*2, 2 DO J = 1, M A(I) = A(I) + B(J) A(I+1) = A(I+1) +B(J) ENDDO		5
4.	Explain in detail with an example why Loop nesting is not always optimal in regard to register reuse	Apply	6
5.	Apply Loop interchange and scalar replacement on following code and find out how many store operations are required for that code. DO I = 2, N DO J = 1, M A (J, I) = A(J, I-1) ENDDO ENDDO	Understand	5

Prepared by: Mr. R.M.Noorullah, Associate professor, CSE
Date : 17September, 2017

HOD, CSE