COURSE INFORMATION

Department/ Faculty:	Electronic Systems Engineering / Malaysia-Japan Int'l Inst. Of Technology	**Page:**	1 of 6
Course code:	SMJE 2173	**Academic Session/Semester:**	2019-2020/ SEM.2
Course name:	DIGITAL SYSTEM DESIGN	**Pre/co requisite (course name and code, if applicable):**	SMJE 1113 DIGITAL ELECTRONICS
Credit hours:	3 (2 + 1) (2 HR LECTURE + 3 HR PRACTICAL)		

Course synopsis	This course introduces design methods to construct digital systems, including combinational and sequential circuits, by means of Hardware Descriptive Language (HDL) specifically Verilog. Areas of topics include: (1) Computer-Aided Design (CAD) tools for design, (2) Verilog Hardware Description Languages (HDL) for simulation and synthesis, and (3) state machine specification, design, and simulation. Principles and fundamentals of digital electronics (SMJE 1113) will be reviewed briefly in refresher session. In this course, some of the important features of HDL will be examined. The course will enable students to design, simulate, model and synthesize digital logic circuits. The dataflow, structural, and behavioral modelling techniques will be discussed and how they are used to design combinational and sequential circuits. Hand-on experience is gained by implementing logic circuits on FPGA development board through weekly lab practices and design project.			
Course coordinator (if applicable)	AP Dr Ooi Chia Yee			
Course lecturer(s)	**Name**	**Office**	**Contact no.**	**E-mail**
	Prof.Dr. Koichiro Mashiko	09.41.01	2203-1320	k.mashiko@utm.my

Mapping of the Course Learning Outcomes (CLO) to the Programme Learning Outcomes (PLO), Teaching & Learning (T&L) methods and Assessment methods:

No.	CLO	PLO	Weight (%)	*Taxonomies / **generic skills*	T&L methods	***Assessment methods
CO1	Apply principles and methodologies of digital system design at registertransfer-level (RTL) and gate-level, including both combinational and sequential circuits.	PLO1 (KW1)	20	C1, C2 / SCH1	Active Learning	Quiz Test Final Exam
CO2	Design digital system that can function correctly on FPGA by	PLO3 (THDS)	30	C3 / TH1 - 3	Active Learning Collaborative Learning	Test, Final exam Lab Practice

Prepared by: Name: Ooi Chia Yee / Koichiro Mashiko Signature: Date: 31 Jan. 2020	Certified by: Name: Dr. Mohd. Ibrahim Shapiai @ Abd. Razak Signature: Date:

	using Hardware Descriptive Language (HDL).				Cooperative Learning	Project
CO3	Manipulate (circuit), synthesis and utilize simulation tools to solve design problems.	PLO5 (THPA)	5	C5 / TH	Cooperative Learning Collaborative Learning	Lab Practice
CO4	Work collaboratively to design and complete a sizable digital system implemented on FPGA.	PLO3- (THDS) PLO9 – (CS) PLO10 –(TW)	(15%) 5 5 5	C6 / TH, CS,TW	Collaborative Learning	PR
Refer *Taxonomies of Learning and **UTM's Graduate Attributes, where applicable for measurement of outcomes achievement ***T – Test; Q – Quiz; HW – Homework; PR – Project; Pr – Presentation; F – Final Exam etc.						

Details on Innovative T&L practices:

No.	Type	Implementation
1.	Lecture	2 hrs/ week x 14 week = 28 hrs
2.	Lab & Project - Practical	3 hrs/ week x 14 week = 42 hrs
3	Continuous Assessment	7.5 hrs
4.	Final Assessment	2.5 hrs
5.	Revision	10 hours
6.	Assessment Preparations	30 hours
	TOTAL	120 hrs

Weekly Schedule:

Week 1	Introduction to Digital Logic Construct Using Verilog - Introduction to HDL and Verilog. Hierarchical design. Verilog constructs – functional, behavioural and structural modelling. Basic Logic Design in Verilog (functional and behavioural) in assign statement and sequential statement: MULTIPLEXER and applications – Include review on digital logic principles of MUX in discrete logics.

Week 2	**Lab Practice Week 2**: Introduction to Quartus® II, FPGA and simple Verilog coding. Constructing MULTIPLEXER in Verilog **Learning Topics:** Basic Logic Design in Verilog (functional and behavioural) in assign statement and sequential statement: DECODERS, TRI-STATE BUFFER, and APPLICATIONS. Arithmetic Circuits: Representation of negative number: sign-and-magnitude, 1's complement, 2's complement 2's complement addition and subtraction. Adder/ Subtractor circuits. Introduction to critical path delay. Examples of critical path delay analysis for adder circuits.
Week 3	**Lab Practice Week 3**: Designing for an application Designing a 4-bit binary input to output a 2-digit decimal display
	Objective: 1) Able to construct and understand the functionality of a simple 2-by-4 decoder 2) Able to construct in Verilog a 7-segment decoder. 3) Able to construct a Verilog module of a simple comparator. 4) Able to integrate all previous modules into a new design project – a BCD converter with a singledigit decimal display **Learning topics:** Comparator and Arithmetic Logic Unit (ALU).
Week 4	**Lab Practice Week 4:** Designing for an application Designing a digit BCD adder 1) Able to construct a 4-bit full adder circuit on FPGA 2) Able to integrate a 4-bit full adder circuit into a new design application – a 2-digit BCD adder with 4-digit decimal display Construct an 8-bit ALU Objective: 1) Able to construct a pre-specified bit-slice design of an ALU. 2) Able to expand the ALU to be 8-bit wide. **Learning topics:** Basic Sequential Circuits Design using Verilog: Sequential statements Blocking vs.Unblocking statements Latches Master-slave Flip-flops timing and clock trigger. Memory

Week 5	**Lab Practice Week 5:** Latches and Flip-flops Constructing latches and flip-flops circuits Objective: 1) Able to construct and simulate in Quartus®II a S-R latch in a 4-input LUT design by means of *assign* statement and structural modeling. 2) Able to construct and simulate a gated-D latch by 'instantiating' a previously built S-R latch 3) Able to construct and perform a master-slave D-flip flop on FPGA by instantiating a previously built gated D-latch. 4) Able to construct a gated D-latch and flip-flop in sequential statements **Learning topics:** **Sequential Circuits Design in Verilog**: Registers, Counters
Week 6	**Lab Practice Week 6:** Slow-clock and sequential logic circuit applications Objective: 1) Able to construct a simple 4-bit binary counter 2) Able to construct variable-size counter by using *parameter* declaration 3) Able to build a 1Hz timer circuit from an on-board 50 MHz crystal oscillator 4) Able to construct a timed circuit – a 3-digit BCD counter **Test 1: Include Week 1 to Week 5 topics and material**
Week 7	**Finite State Machine Design using Verilog**
	Finite state machine – concept and structure. Moore and Mealy machines. Design procedure. Demonstrate examples of finite state machines.
Week 8	**Mid-Semester Break**
Week 9	**Lab Practice Week 9:** Project-based work **Learning topics:** **Finite State Machine Design using Verilog** Demonstrate examples of finite state machines. Re-visit critical path delay analysis. Setup time and hold time.
Week 10	**Lab Practice Week 10:** Project-based work **Learning topics:** **Verification:** Testbench

Week 11	**Lab Practice Week 11: Project-based work** **Learning topics:** **Verification:** Testbench
Week 12	**Lab Practice Week 12:** Project-based work **Test 2: Sequential Logic Circuits**
Week 13	**Group reporting on project progress** **Lab Practice Week 13:** Project-based work
Week 14	**Group reporting on project progress** **Lab Practice Week 14:** Project-based work
Week 15	**PRESENTATIONS** **Lab Practice Week 15:** Project-based work – PRESENTATIONS

Transferable skills (generic skills learned in course of study which can be useful and utilised in other settings):

Student learning time (SLT) details:

Distribution of student Learning Time (SLT) Course content outline		Teaching and Learning Activities		TOTAL SLT
	Guided Learning (Face-to-Face) T(tutorial), P(Practical), O(??)	Guided Learning Non Face-to-Face	Independent Learning Non Face-to-Face	

CLO - PLO	**Lecture**	**T**	**P**	**O**			
CO1-PO1, [Principles of RTL] Lecture only	10					10	**20**
CO2 – PO3, [HDL programming] Lecture, Lab, Project	18		20			10	**58**
CO3 – PO5 [Quartus2&FPGA] Lab, Project			10			10	**38**
CO4 – PO10 Project			12			10	**9**

Total SLT	**28**		42			**40**	**110**

Continuous Assessment		PLO	Percentage	Total SLT
1	Quiz	1	5%	0.5 hr
2	Test 1	1	10%	1.0 hr
3	Test 2	1, 3	15%	1.0 hr
4	Lab Work/ Practice	2, 3	5%	3.0 hr
5	Project – (Report & Project demo)	2,3,4	15%	2.0 hr
	Total SLT (Assessment)		**50%**	**7.5 hr**
Final Assessment			Percentage	Total SLT
1	Final Exam		50%	2.5 hrs
	Total SLT (Assessment)			**2.5 hrs**
Grand Total SLT				**120**

Special requirement to deliver the course (e.g: software, nursery, computer lab, simulation room):

1) A computer system installed with and FPGA EDA GUI – Altera® Quartus II , Xilinx® ISE or Xilinx® Vivado
2) FPGA development board at least Altera© DE2-115

Learning resources:

Text book (if applicable)

1. Khalil Mohamed Hani, "RTL Design of Digital Systems with Verilog HDL" Univ. Teknologi. Malaysia, 2017.
2. Stephen Brown and Zvonko Vranesic, **"Fundamentals of Digital Logic with Verilog Design"**, *3rd Edition* , Mc GrawHill, 2014

Main references

1. Peter J. Ashenden, *Digital System: An Embedded System Approach Using Verilog*, Morgan Kaufmann Publishers, 2008.

Additional references

-

Online

-

Academic honesty and plagiarism:

Academic honesty and integrity is fundamental in any education institution and a principle to good academic work. We, in the university, view cheating as an unethical conduct. Cheating, copying and plagiarizing are serious violations to principles of academic honesty. Students who cheat consider the work they produced as theirs and undermine the effort and hard work of people who actually produce the original work. Strong penalty shall be imposed to students who commit such offense.

- All programming codes, documentation, lab reports and assignments must be from the student's own effort.
- In programming, when a student copies code from an external source regardless whether he is copying a snippet of code or an entire module, the student should credit the source. The student should explicitly cite the work.
- When a student copies the code and adapt it, the student should still credit the source. The student was not the original developer of the code
- Students are expected to cite, paraphrase, disclose on any similarity found on their work with an original source.
- In the case of a group work, each member of the group is held equally accountable and responsible for any kind of cheating found in the work which the group presented to the lecturer.
- Academic dishonesty conducted in this course will be penalized and the penalties are , but not limited to, as the following:
  (i) The assignment/ report/ lab work/ project will not be graded
  (ii) The work will be omitted and student have the opportunity to deliver a new work but at a reduced grade.
  (i) Every member of the group (in group work) shall be penalized equally.
- Those who invest the time working through the problem sets are better prepared to answer exam questions that call for conceptual thinking. And those who copy and cheat are far more likely to fail in the subject than those who did not copy.

Other additional information (Course policy, any specific instruction etc.):

Disclaimer: