



RENCANA PEMBELAJARAN SEMESTER (RPS) UNIVERSITAS DIPONEGORO

SPMI- UNDIP	RPS	S1.SK-FT-UNDIP	071
--------------------	------------	-----------------------	------------

Revisi ke	1
Tanggal	5 Juli 2022
Dikaji Ulang Oleh	Ketua Program Studi Teknik Komputer
Dikendalikan Oleh	GPM Teknik Komputer
Disetujui Oleh	Dekan Fakultas Teknik

UNIVERSITAS DIPONEGORO		SPMI-UNDIP/RPS/S1.SK-FT-UNDIP/071	Disetujui Oleh Dekan Fak. Teknik
Revisi ke 1	Tanggal 5 Juli 2022	Rencana Pembelajaran Semester	



UNIVERSITAS DIPONEGORO
FAKULTAS TEKNIK
DEPARTEMEN TEKNIK KOMPUTER
PROGRAM SARJANA

SPMI-
UNDIP/RPS/S1.SK-
FT-UNDIP/071

**RENCANA PEMBELAJARAN
SEMESTER**

MATA KULIAH (MK)	KODE	RUMPUN MK	BOBOT		SEMESTER	TANGGAL PENYUSUNAN
Perancangan System on Chip(SoC)	PTSK6751	Pilihan	T=2	P=0	Ganjil	5 Juli 2022
OTORISASI/PENGESAH AN	DOSEN PENGEMBANG RPS		Koordinator RMK		KaPRODI	
	Eko Didik Widiyanto, ST, MT					
CAPAIAN PEMBELAJARAN	CPL yang dibebankan pada MK					
	CPL 3	Memiliki pemahaman keilmuan dan penguasaan keterampilan di bidang teknik komputer, meliputi sistem tertanam dan robotika, jaringan dan keamanan komputer, rekayasa perangkat lunak, multimedia, game, dan kecerdasan buatan yang ditopang oleh profesionalitas, pengetahuan sains dasar dan rekayasa yang kuat.				
	CPMK (Capaian Pembelajaran Mata Kuliah)					
	CPMK 3-1	Mampu mendesain purwarupa sistem standalone dengan konsep waktu nyata untuk perancangan sistem kontrol dan monitoring				
	CPMK 3-2	Mampu mendesain purwarupa sistem terdistribusi dengan konsep waktu nyata untuk perancangan sistem kontrol dan monitoring				
Deskripsi Singkat	Kuliah ini mempelajari dasar-dasar pengembangan system-on-chip (SoC) meliputi pengenalan perangkat desain SoC, pemodelan sistem, pengantar TLM dan pemetaan perangkat keras, model-model komputasi, metodologi desain SoC, H/W co-design, sintesis high-level, sistem penjadwalan, alokasi sumber daya, emulasi, verifikasi, dan integrasi sistem SoC.					
Bahan Kajian Materi Pembelajaran	1. Pendahuluan SoC 2. Perangkat Desain SoC 3. Pemodelan System 4. Pengantar TLM 5. Model model Komputasi 6. Metodologi Desain SoC 7. HW/SW Co Desain 8. Sintesis 9. Sistem Penjadwalan 10. Alokasi Sumber Daya 11. Emulasi dan FPGA 12. Verifikasi Desain SoC 13. Pengujian SoC 14. Proyek					

Pustaka		Utama: 1. Keating, M. (2011). The Simple art of SoC design. Springer. ISBN 9781441985859. 2. OSCI. SystemC tutorials and whitepapers. Download from OSCI http://accellera.org/community/systemc or copy from course web site. 3. Ghenassia, F. (2010). Transaction-level modeling with SystemC: TLM concepts and applications for embedded systems. Springer. 4. Eisner, C. & Fisman, D. (2006). A practical introduction to PSL. Springer (Series on Integrated Circuits and Systems). 5. Foster, H.D. & Krolnik, A.C. (2008). Creating assertion-based IP. Springer (Series on Integrated Circuits and Systems). 6. Grotker, T., Liao, S., Martin, G. & Swan, S. (2002). System design with SystemC. Springer. 7. Wolf, W. (2009). Modern VLSI design (System-on-chip design). Pearson Education (4th ed.). 8. Sumber lain: paper ilmiah, website project					
Pengampu		Team Pengajar Perancangan System On Chip					
Prasyarat		-					
Media Pembelajaran		Papan Tulis, LCD Projector, Laptop, dan Power Point					
Mg ke-	Sub-CPMK (sebagai kemampuan akhir yang diharapkan)	Penilaian		Bentuk pembelajaran; Metode Pembelajaran; Penugasan; [Estimasi Waktu]		Materi Pembelajaran	Bobot (%)
		Indikator	Kriteria dan Bentuk	Tatap Muka/Luring	Daring		
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
1	Mampu menjelaskan tentang SoC dan alur desain SoC paling sedikit 80% tepat.	• Mampu menjelaskan konsep SoC • Mampu menjelaskan perbedaan pendekatan SoC dengan lainnya • Mampu memahami alur desain SoC dan menjelaskannya untuk permasalahan sederhana	Kriteria: Ketepatan sesuai pedoman penilaian Bentuk non-test: Merangkum bahan ajar	Ceramah, Small Group Discussion, Problem Based Learning (TM; 2x (2x50"))		1. Pendahuluan Perancangan System-on-Chip (SoC) 1.1 Pengantar SoC 1.2 Alur desain SoC 1.3 Tujuan dan rencana perkuliahan Deskripsi proyek	

2	Mampu Menjelaskan Lingkup Perangkat Lunak untuk Desain SoC dan menerapkannya di lingkungan pengembangan SoC paling sedikit 80% tepat	• Mampu menjelaskan perangkat lunak pengembangan sistem SoC; • Mampu memahami fitur-fitur lingkungan pengembangan; • Mampu menggunakan perangkat lunak tersebut	Kriteria: Ketepatan sesuai pedoman penilaian Bentuk non-test: Merangkum bahan ajar	Ceramah, Small Group Discussion, Problem Based Learning (TM;2x (2x50"))		2. Perangkat Perangkat Desain SoC 2.1 Lingkungan SystemC 2.2 Arsitektur SystemC	10
3	Mampu menjelaskan dan merancang pemodelan sistem, level abstraksi model, dan pemrograman di level sistem paling sedikit 80% tepat	• Mampu memahami pemodelan sistem dan SystemC • Mampu menjelaskan dan menerapkan level-level abstraksi model • Mampu merancang model di level sistem	Kriteria: Ketepatan sesuai pedoman penilaian Bentuk non-test: Merangkum bahan ajar	Ceramah, Small Group Discussion, Problem Based Learning (TM;2x (2x50"))		3. Pemodelan Sistem dan System-C 3.1 Level abstraksi model 3.2 Pemrograman Level Sistem	

4	Mampu merancang model di level transaksi /TLM paling sedikit 80% tepat	- Mampu memahami tentang sistem - Mampu memahami tipe-tipe pemodelan - Mampu membedakan antara pendekatan desain tradisional dan baru - Mampu memahami TLM dan mengaplikasikannya	Kriteria: Ketepatan sesuai pedoman penilaian Bentuk non-test:	Ceramah, Small Group Discussion, Problem Solving (TM;2x (3x50"))	4. Pengantar TLM dan Pemetaan Perangkat Keras 4.1 Sistem 4.2 Tipe-tipe pemodelan 4.3 Desain tradisional 4.4 Desain baru TLM (Model level transaksi)	10
			Merangkum bahan ajar			
5	Mampu menjelaskan model-model komputasi paling sedikit 80% tepat	- Mampu memahami model-model komputasi - Mampu menjelaskan proses desain - Mampu menjelaskan spesifikasi dan - pemodelan H/W co-design	Kriteria: Ketepatan sesuai pedoman penilaian Bentuk non-test: Merangkum bahan ajar	Ceramah, Small Group Discussion, Problem Based Learning (TM;2x (2x50"))	5. Model-model Komputasi 5.1 Sistem embedded 5.2 Proses desain 5.3 Spesifikasi dan pemodelan H/W co-design 5.4 Model model komputasi	

6	Mampu menjelaskan dan mengaplikasikan metodologi dengan SoC paling sedikit 80% tepat	- Mampu memahami dan menerapkan alur desain SoC - Mampu menjelaskan arsitektur MPSoC - Mampu mengaplikasikan metodologi desain di level sistem meliputi pemodelan, sintesis, dan verifikasi	Kriteria: Ketepatan sesuai pedoman penilaian Bentuk non-test: Merangkum bahan ajar	Ceramah, Small Group Discussion (TM;2x (2x50"))		6. Metodologi Desain SoC 6.1 SoC design flow 6.2 Multiprocessor System-on-Chip (MPSoC) 6.3 Metodologi desain level	10
7	Mampu menjelaskan dan mengaplikasikan HW/SW co-design paling sedikit 80% tepat.	- Mampu menjelaskan desain sistem teraserasi - Mampu menjelaskan analisis performansi - Mampu menjelaskan partitioning dan penjadwalan dalam HW/SW co-design - Mampu menjelaskan perkembangan MPSoC - Mampu menerapkan teknik-teknik HW/SW co-design	Kriteria: Ketepatan sesuai pedoman penilaian Bentuk non-test: Merangkum bahan ajar	Ceramah, Small Group Discussion (TM;2x (2x50"))		7. HW/SW Co-design 7.1 Desain Sistem Teraserasi 7.2 Analisis performansi 7.3 Partitioning 7.4 Penjadwalan 7.5 Perkembangan MPSoC	10
UTS							40

8	Mampu menjelaskan dan mengaplikasikan sintesis di level tinggi paling sedikit 80% tepat.	- Mampu menjelaskan sintesis level tinggi (HLS) - Mampu menyusun spesifikasi perilaku, arsitektur target, dan representasinya - Mampu menjelaskan dan menerapkan alur dan strategi sintesis level tinggi	Kriteria: Ketepatan sesuai pedoman penilaian Bentuk non-test:	Ceramah, Small Group Discussion (TM;2x (2x50")) Ceramah, Small Group Discussion (TM;2x (2x50")) Ceramah, Small Group Discussion (TM;2x (2x50"))		8. Sintesis (High-level) 8.1 Pendahuluan High-level synthesis (HLS) 8.2 Spesifikasi perilaku, arsitektur target, representasi 8.3 Alur sintesis level tinggi 8.4 Strategi sintesis	5
9	Mampu menjelaskan dan menerapkan sistem penjadwalan dalam desain SoC paling sedikit 80% tepat.	- Mampu menjelaskan permasalahan dalam penjadwalan - Mampu membedakan teknik penjadwalan tanpa konstrain dan dengan konstrain - Mampu menerapkan teknik penjadwalan tanpa Konstrain - Mampu menerapkan teknik penjadwalan dengan konstrain sumber dayadan waktu - Mampu menerapkan teknik penjadwalan chaining dan pipelining	Kriteria: Ketepatan sesuai pedoman penilaian Bentuk non-test: Merangkum bahan ajar	Ceramah, Small Group Discussion (TM;2x (2x50"))		9. Sistem Penjadwalan 9.1 Permasalahan Penjadwalan 9.2 Penjadwalan tanpa konstrain 9.3 Penjadwalan dengan konstrain sumber daya 9.4 Penjadwalan	5

						dengan konstrain waktu Penjadwalan lanjut:	
10	Mampu menjelaskan dan menerapkan alokasi sumber daya dalam desain SoC paling sedikit 80% tepat.	• Mampu menjelaskan dan mendesain alokasi dan penggunaan bersama sumber daya • Mampu menjelaskan dan menerapkan penggunaan bersama unit fungsional, register, dan bus • Mampu menjelaskan dan menerapkan ekstensi modul • Mampu melakukan merancang dan mensintesis datapath dan kontrol	Kriteria: Ketepatan sesuai pedoman penilaian	Ceramah, Small Group Discussion (TM; 2x (2x50"))		10. Alokasi Sumber Daya 10.1 Alokasi dan penggunaan bersama sumber daya 10.2 Penggunaan bersama unit fungsional 10.3 Penggunaan bersama register 10.4 Penggunaan bersama bus 10.5 Ekstensi modul 10.6 Sintesis datapath dan kontrol	10

11	Mampu menjelaskan tentang emulasi dan FPGA paling sedikit 80% tepat.	- Mampu menjelaskan tentang emulasi dan menerapkan tekniknya - Mampu menjelaskan tentang perkembangan FPGA, CPLD, dan ASIC	Kriteria: Ketepatan sesuai pedoman penilaian Bentuk non-test:	Ceramah, Small Group Discussion (TM;2x (2x50"))		11. Emulasi dan FPGA 11.1Perkembangan FPGA, CPLD, dan ASIC 11.2Emulasi FPGA	5
			Merangkum bahan ajar				
12	Mampu menjelaskan dan menerapkan verifikasi desain SoC paling sedikit 80%tepat	- Mampu menjelaskan dan menerapkan teknik simulasi dalam desain SoC - Mampu menjelaskan dan menerapkan analisis formal	Kriteria: Ketepatan sesuai pedoman penilaian Bentuk non-test: Merangkum bahan ajar	· Ceramah Context ual Instruction (TM;2x (2x50"))		12. Verifikasi desain SoC 13.1 Teknik berbasis simulasi 13.2 Analisis formal 13.3 Verifikasi software tertanam	10
13	Mampu menjelaskan dan menerapkan pengujian SoC paling sedikit 80% tepat.	- Mampu menjelaskan fiturtestabilitas SoC dengan boundary scan danstandar P1500 - Mampu menjelaskan cost-cost dalam pengujian SoC - Mampu menjelaskan dan menerapkan built-in self testing - Mampu menjelaskan dan menerapkan deteksi error - dan sistem fault-tolerant	Kriteria: Ketepatan sesuai pedoman penilaian Bentuk non-test: Merangkum bahan ajar	Ceramah Diskusi (TM;2x (3x50"))		13. Pengujian SoC 13.1 Fitur testabilitas SoC: boundar y scan, standar P1500 13.2 Cost pengujian SoC 13.3 Pengujian mandiri built- in 13.4 Toleransi defect 13.5 Deteksi error dan	5

						toleransi fault	
14	Mampu merancang, menganalisis dan menguji SoC paling sedikit 80% tepat.	Mampu menerapkan teknik-teknik dalam metodologi dalam mengembangkan sistem SoC sederhana	Kriteria: Ketepatan sesuai pedoman penilaian Bentuk non-test: Merangkum bahan ajar	Context ual Instruction Project-based learning (TM;2x (2x50"))		Projek perancangan SoC	5
UAS							40

