1. Services

1.1 Consulting Services

Services offered to a unique customer to deliver modified or completed electronic designs, including semiconductor or Semiconductor Intellectual Property (SIP) products, printed circuit board (PCB), module, system designs, libraries, verification, retargeting, etc. Also includes services offered to a unique customer to provide advice and support to help transform, optimize, and port design flows and methodologies.

1.2 Custom Tool Development & Design Environments and Product Data Management (PDM) and Component Information Systems (CIS) Services

Services offered to a unique customer to customize design tools and their environments, product data management systems, and component information management.

1.3 Training Services

Educational products and/or services (excluding university programs/training) offered to a broad set of customers, relating to electronic design, electronic design methodology, electronic design languages, and/or the use of electronic design automation tools.

1.4 Other Services

Services offered to a broad set of customers, such as product installation and field application support.

2. Computer-Aided Engineering (CAE) Tools

2.1 Electronic System Level (ESL) Design, Synthesis and Verification

Software tools that design, model, simulate, create and/or analyze the functionality and performance of system-level designs (including elements of hardware and software), as well as the interaction of the hardware and software elements of these designs.

2.1.1 ESL Design

Software tools used to model, develop, debug, analyze, simulate and visualize the functional behavior of a design (or parts of a design) modeled at a high-level of abstraction, either untimed or containing timing specifications, as well as software tools that enable the designer to analyze architectural characteristics of system-level designs such as power, area, verifiability, bus optimization, and memory utilization (including elements of hardware and software). This category also includes software tools used to assemble a collection of hardware and software IP blocks into a system (including platform-based design tools). This category does not include system-level verification tools (see 2.1.3.1 ESL Verification).

2.1.2 ESL Synthesis: Software tools that transpose a C/C++, SystemC or similar high level input design abstraction and implement the design in an RTL-Level design

description output. Tools in this category at their essence include a "scheduler" which is responsible for automatically assigning (scheduling) operations over clock cycles.

2.1.3 ESL Verification and ESL Virtual Prototyping: Software tools that simulate and verify the functionality and/or performance of a system-level design, including the verification of the interaction between the design's hardware and software elements. This includes tools and interfaces that link hardware/software simulation and analysis to software debuggers and logic simulators, and tools that provide the simulated performance of a design (or part of a design) to a target user of the functionality of the design. (I.E., a virtual prototype of a hardware and firmware design for use by a software designer for software application development and/or verification). This category only includes tools that are specific to system-level verification and which are not covered in other categories, such as logic verification.

2.2 Design Entry Tools (Textual and Graphical)

Software tools that support design entry via hardware description languages, Boolean equations, high-level graphical methods, schematic editors used for integrated circuits, or a combination of these methods. High-level graphical methods may also include a schematic editor if sold in a bundled package. This category includes analog schematic entry, but not generic drawing packages.

2.3 Logic Verification

2.3.1 RTL Simulation: Simulators and integrated simulation tool suites used to verify logic designs using one or more RTL languages (i.e. VHDL, Verilog, SystemVerilog).

2.3.2 Dynamic Verification and Auxiliary RTL Simulation: Software tools that connect to RTL simulators to assist in the verification of RTL-based designs.

2.3.3 Hardware-Assisted Verification: Hardware-based design verification tools, including emulation and acceleration.

2.3.4 Other Logic Verification and Simulation Tools (Including Gate-Level Simulators): Any other Logic Verification and Simulation Tools not listed above. This includes Gate-Level Simulators and non-simulation-based techniques to assist with the verification of digital designs, such as linters, coverage checkers, and assertion-based verification point tools. Note: Equivalence checking tools are reported separately under formal verification. Additionally, static timing analysis is reported separately under analysis tools. Excludes formal-based tools reported in 2.5.2.

2.4 Analog & Mixed Signal Simulators

2.4.1 Analog & Mixed Signal Simulators: Software tools that simulate analog or mixed (analog and digital) signals. Includes interface packages for linking analog and digital signals.

2.4.2 Library Characterization: Software tools used to characterize standard cells, I/O, macro or memory SIP and generate the timing, noise and power libraries required for analysis, synthesis and physical implementation tools.

2.4.3 RF Simulators: Software tools that simulate both at the circuit-level and/or system-level for RF/high-frequency and/or microwave designs. This category includes tools that perform linear frequency-domain simulation, harmonic-balance based simulators, circuit envelope simulators, RF/microwave transient/convolution simulators, and any RF/communication system-level simulators. It does not include tools defined in 2.6.5 as Analog and High-Frequency Analysis Tools.

2.4.4 EM Solvers, or Planar & 3D EM Solvers: Software tools that model and analyze the electrical characteristics and/or performance of physical geometries and/or structures, both 2-dimensional (2D, 2.5D) and/or 3-dimensional (3D), through electromagnetic techniques. This category excludes software tools identified as Parasitic Extraction Tools defined in 4.4.

2.5 Formal Verification

2.5.1 Equivalency Checking: Software tools that use formal techniques to verify the functional equivalence of a design as it is transformed from one stage of the design process to the next. Includes RTL-to-RTL, RTL-to-Gate, Gate-to-Gate, etc.

2.5.2 Property Checking: Software tools that use formal analysis techniques - statically or in conjunction with other means such as simulation - to verify design properties such as assertions, assumptions, constraints, etc. used to define proper functional behavior of a design, which can be either user-specified or automatically extracted.

2.6 Analysis Tools

The tools in this category may work at any level of abstraction: behavioral, register-transfer-level (RTL), gate-level, or the physical layout of a device or system.

2.6.1 IC/ASIC Static Timing and Signal Integrity Analysis: Software tools that calculate delays (delay calculators) or detect timing violations in a digital design by checking the clock frequency with appropriate gate-level and interconnect delay models. Also includes software tools used to analyze electrical signal behavior in wiring networks contained in integrated circuits (ICs), including cross-talk and substrate noise analysis.

2.6.2 IC/ASIC Power Analysis and Optimization: Software tools that analyze, optimize, or diagnose power consumption and IR drop problems, or provide automatic power reduction in electronic circuits.

2.6.3 IC/ASIC Transistor-Level Simulation and Analysis: High-capacity circuit simulation tools designed specifically for timing and/or power analysis, that accept a SPICE netlist for input, and that handle capacities of a million or more transistors.

2.6.4 Analog and High-Frequency IC/ASIC Analysis: An environment and its sub-tools dedicated to setting up and running analog, RF, and/or mixed-signal simulators and analyzing the results that are produced in either a manual or automated manner. Does not include simulation or analysis tools defined elsewhere.

2.6.5 Other IC/ASIC-Related Analysis: Software products used to analyze electrical, thermal, EMC, power, and timing related to IC wiring networks. This category also includes tools used in conjunction with analysis, such as floorplanning, interconnect tracing, and topology extraction.

2.7 Design-for-Test and Test Automation Tools

2.7.1 ATPG: Automatic Test Pattern Generation for full-scan, partial-scan, and non-scan designs.

2.7.2 BIST: Insertion of circuitry or IP blocks to perform a Built-in Self-Test function for memories, random logic, mixed signal, etc.

2.7.3 Scan: Insertion of either internal scan circuitry to support ATPG or BIST, or IEEE 1149.1, 1149.6, or 1149.7 boundary scan circuitry.

2.7.4 Fault Simulation and Other Test: Fault simulation/grading of functional vectors. Also includes methods or IP for test reuse, scan wrappers, pattern mapping, etc.

2.8 Synthesis

Software tools that read a high-level electronic design description and implement it at a lower level of abstraction. Logic synthesis tools typically implement a design down to the gate level, and provide for logic optimization and library retargeting using a custom, ASIC, or FPGA design implementation.

2.9 Other CAE Hardware & Software

Any other CAE hardware or software tool not listed above.

3. PCB & MCM Layout Tools

3.1 PCB Schematic Entry

Printed Circuit Board (PCB) software products commonly known as schematic editors or schematic capture tools. This category also includes analog schematic entry, but not generic drawing packages.

3.2 PCB Analysis

Software products used to analyze electrical, thermal, EMC, power, and timing related to wiring networks in PCBs other PCB-related designs. This category also includes tools used in conjunction with analysis, such as PCB floorplanning, interconnect tracing, and topology extraction.

3.3 IC Package Analysis

Software products used to analyze electrical, thermal, EMC, power, and timing related to wiring networks in IC Packages, multi-chip packages or MCMs. This category also includes tools used in conjunction with analysis, such as floorplanning, interconnect tracing, and topology extraction.

3.4 Other System Interconnect Analysis

Software products used to analyze electrical, thermal, EMC, power, and timing related to wiring networks other than PCBs or IC Packages, including cables, harnesses, connectors, sockets, optics. This category also includes tools used in conjunction with analysis, such as floorplanning, interconnect tracing, and topology extraction.

3.5 PCB Computer-Aided Manufacturing

Software tools used to interface design with manufacturing for PCB design.

3.6 PCB Physical Design

Physical design tools for the placement of physical components and/or the routing of interconnect signal traces on printed circuit board (PCB) assemblies. At a minimum, a PCB layout tool must have the capability to determine the placement of components or to route interconnect wiring. A PCB layout tool may include design rule checking, photoplotting output, provided that it handles layout.

3.7 IC Package Physical Design

Physical design tools for the placement of physical components and/or the routing of interconnect signal traces on IC Package or multi-chip package assemblies. At a minimum, a layout tool must have the capability to determine the placement of components or to route interconnect wiring. A layout tool may include design rule checking, mask outputs, and interfaces to manufacturing, provided that it handles layout.

3.8 Other Physical Design

Physical design tools for fabrics other than PCB and Packaging such as cables, harnesses, connectors, sockets, optics. This may involve placement of physical components and/or the routing of interconnect. May also include standalone design rule checking, mask outputs, and interfaces to manufacturing.

3.9 Library & Design Data Management

Descriptions and management of design elements used for designing physical interconnect systems. This category includes component models for simulation or

analysis, symbols, component information systems, library development tools, library management tools, and design libraries. It does not include semiconductor IP (SIP), blocks, or libraries used in IC design.

3.10 Other PCB & MCM Hardware & Software

Any other PCB & MCM hardware or software tool not listed above.

4. IC Physical Design & Verification Tools

4.1 Physical Implementation

Software tools that automatically perform the placement and routing of circuits on an integrated circuit (IC) or application-specific integrated circuit (ASIC). Includes tools for designing gate arrays, embedded arrays, standard cells, and irregularly-sized macro- or mega-cell blocks. Also includes software tools that analyze timing, size, power dissipation, and routability before detailed physical layout. These include tools that provide estimations of interconnect resistance, capacitance, and/or inductance. May also include tools that perform synthesis and placement (but not routing), solely for the purpose of creating high-level graphical depictions of the topology of an integrated circuit layout.

4.2 IC Full Custom Layout

Software tools for hand-crafted full-custom ICs. Includes polygon editors, symbolic editors, and compactors.

4.3 IC Layout Verification

Software tools that verify that the layout topology of circuits which has undergone placement, routing, and compaction does not violate any fabrication process rules, i.e. design rule checkers (DRC). Includes electrical rule checkers (ERC), which verify that no electrical rule violations have occurred, and layout-versus-schematic (LVS) checkers, which verify that the physical implementation of the design matches its logical implementation.

4.4 Parasitic Extraction

Software tools that translate IC layout data into networks of electrical circuit elements (transistors, resistors, and capacitors) and parasitic elements (interconnect capacitance, resistance, and inductance). These tools are used to model the timing, power, and signal behavior of the IC design. Includes network reduction tools and two-dimensional and three-dimensional field solvers.

4.5 RET EDA (Including OPC and PSM)

EDA software used for modifying full-chip production designs on an existing defined production process to enhance resolution, process-window, and ultimately yield. Reticle Enhancement Technology (RET) software supports the full-layer layout geometry manipulation and empirically-modeled simulation necessary for a range of RET layout

modifications such as Optical Proximity Correction (OPC), Strong Phase Shift Mask (PSM), Attenuated PSM, Scattering Bars, and support for illumination patterns such as Quadrupole and Annular. Rule-based OPC tools are reported in the IC Layout Verification Tools (4.3) category. Model-based OPC tools are reported in this category.

4.6 Technology CAD (TCAD)

Software used for simulating, exploring, analyzing and optimizing device, process, and/or optical parameters during the process of researching and developing a new semiconductor process. These tools are never used for production processing of complete IC designs, and are typically used only on very small test structures due to computational constraints and the complexity of simulation setup and calibration. The input parameters to simulation are numerous first-principle-type physical parameters rather than empirical, summary-type parameters. These tools are sometimes used for calibrating full-chip production software tools.

4.7 Mask Data Prep

Software used for modifying full-chip production designs from the physical designer's DRC-clean layout into the various modified layers required for IC manufacturing (geometry processing), and the software that is used in most, but not all, cases to translate the design data from the EDA-standard formats into proprietary mask-writer (fracturing) and mask-inspection formats, including proprietary machine formats. This category also includes software used for frame generation, reticle layout and floorplanning (reticle organization), wafer floorplanning, and optimization.

4.8 Yield Enhancement

Software tools that modify a physical layout to avoid manufacturing process vulnerabilities and improve chip yield. Excludes RET & OPC tools reported in 4.5.

4.9 Other IC/ASIC and FPGA Physical Design and Verification Tools

Any tools for IC/ASIC and FPGA physical design and verification and yield optimization that are not listed above.

5. Semiconductor Intellectual Property (SIP)

5.1 SIP-Related Tools

Software tools used in the generation, creation, packaging, and management of SIP. Also includes tools for the development of module generators. Examples include: module generators (AKA "target compilers"), used exclusively to generate physical SIP blocks from regular parameterizable physical structures that are based on a fixed set of base leaf cells; development Tools for Module Generators, used for developing module generators, for porting or compacting physical libraries, and automating the characterization, documentation, quality assurance, and generation of all the EDA views necessary to use the libraries, as well as IC/ASIC library porting tools. Also includes SIP Creation and packaging tools used for authoring new reusable SIP blocks or for converting existing

SIP blocks from a non-portable to a portable form and for createing functional and timing "views" of an SIP block, as well as protected (encrypted) versions of Proprietary SIP and/or Standards-Based SIP. Additional examples include tools for SIP database management, delivery, and publishing.

5.2 Macrocells and Cores

5.2.1 Logic Libraries/Standard Cells: The class of building blocks or elements used to assemble or compile a virtual component into a particular target process. The library provides a physical representation of the logic and functional elements of the design. Includes standard cells, memory, arithmetic/mathematic, interface, GPIO, etc.

5.2.2 Memory Elements: Any storage element, from circular buffers and memory cells up to complete memory blocks. Includes static, dynamic, and non-volatile memory.

5.2.3 Analog and Mixed Signal: An IP block or virtual component with circuitry that provides analog functionality, usually requiring additional considerations over standard digital SIP. Includes: virtual components and interfaces for wireless radio-frequency physical media interfaces; A/D, D/A, comparators, amplifiers, detectors, pulse compression, sources, switches, PLL/VCO, reference/regulators, pulse-width modulators, and other components; filters, couplers, doppler, target and clutter, mixers, and multipliers/dividers.

5.2.4 Interface/Peripheral Cores: Interfaces and peripherals (in the form of software or RTL) that conform to recognized standards, or which perform standard functions such as timers or keyboard controllers.

5.2.4.1 xDSL

5.2.4.2 PCI Controllers: Includes PCI Controller Cores (Soft RTL, Firm, Hard), PCI Controller PHY, and PCI Stacks and Drivers.

5.2.4.3 USB Controllers: Includes USB Controller Cores (Soft RTL, Firm, Hard), USB Controller PHY, and USB Stacks and Drivers.

5.2.4.4 PCMCIA

5.2.4.5 IEEE 1284

5.2.4.6 IEEE 1394

5.2.4.7 SONET

5.2.4.8 Ethernet (IEEE802.x) Controllers: Includes Ethernet Controller Cores (Soft RTL, Firm, Hard), Ethernet Controller PHY, and Ethernet Stacks and Drivers.

5.2.4.9 Bluetooth

5.2.4.10 Rapid I/O

5.2.4.11 Infiniband

5.2.4.12 Fiberchannel

5.2.4.13 SPI-4

5.2.4.14 DDR Controllers: Includes DDR Controller Cores, DDR Controller PHY, and DDR Stacks and Drivers.

5.2.4.15 SATA Controllers: Includes SATA Controller Cores, SATA Controller PHY, and SATA Stacks and Drivers.

5.2.4.16 PATA Controllers: Includes PATA Controller Cores, PATA Controller PHY, and PATA Stacks and Drivers.

5.2.4.17 Other Interface/Peripherals: Receivers/transmitters, timers, DMA controllers, keyboard controllers, HDMI, and others.

5.2.5 CODEC/Encryption: All blocks used to encode or encrypt information for transmission purposes.

5.2.5.1 Encoders/Decoders: Includes MPEG, Viterbi, Reed-Solomon, Turbo-coding, and other Encoders/Decoders.

5.2.5.2 Encryption/Decryption

5.2.5.3 Modulators/Demodulators

5.2.5.4 Other CODEC/Encryption (Error Correction/Detection)

5.2.6 Graphics, Imaging, and Audio: Blocks that perform graphic-, audio-, or image-processing functions to recognized standards or provide functions such as edge detection.

5.2.7 Processors: Any virtual component or block that can run software to perform a task.

5.2.7.1 Control plane processor SIP: All controllers and CPUs, including both real-time controllers and full application processors. The cores serve as the main

programmer visible processing engine(s) on a device. Includes "multi-core" SMP processors.

5.2.7.2 Dataplane processor SIP: Communications DSPs, general purpose DSPs, media DSPs (audio, video, or imaging), graphics processors, network or packet processors (NPUs), security processors, and real-time embedded controllers (only when the device contains another main controller/CPU in the control plane). All devices must be software programmable - thus excluding "hardware accelerators" that do not have their own standalone software program stream.

5.2.8 Subsystems: Blocks that are made by combining more than one sub-block to form a subsystem or platform, including software and hardware blocks.

5.2.9 Test Functions: Virtual components that relate exclusively to test functions. Includes debug, self-test, and others.

5.2.10 DSP Functions: Virtual components that perform digital signal processing functions. This category pertains to dedicated hardware blocks that perform specific DSP algorithms or routines. Includes accelerators, building blocks, correlators, filters, transformers, and others.

5.2.11 Other Macrocells and Cores: Any new functional class that arises is put in this category until a separate class is created.

5.3 Verification SIP

SIP for the verification of other hardware IP or subsystems, including models, monitors, test suites, testbenches, assertions, and checkers.

5.4 Embedded Software

5.4.1 Real-Time Operating Systems

5.4.2 Stacks & Drivers: Communications protocols and other hardware-dependent software not associated with interface controllers in 5.2.6.

5.4.3 Applications: DSP or other real-time application software.

5.4.4 Other Embedded Software