1. Выберите то, что **является** преимуществами **гибкой** логики по сравнению с **жесткой** логикой.
   1. Низкая избыточность чипа.
   2. **Высокая скорость разработки чипа.**
   3. **Возможность организации схемы решения широко перечня задач.**
   4. Высокая скорость работы схемы.
   5. Низкое энергопотребление схемы
   6. Низкая цена производства схемы.
   7. Высокая цена ошибки решения задачи.
   8. Меньшие габариты чипа.
   9. **Возможность интуитивно-понятной и простой реализации сложных логических функций.**
   10. **Низкий «порог знаний для входа» - освоения системы.**
2. Выберите то, что **является** преимуществами гибкой **последовательной** логики по сравнению с гибкой **параллельной** логикой.
   1. Высокая скорость работы спроектированного устройства.
   2. **Простота разработки программного обеспечения с точки зрения «порога знаний для входа».**
   3. Высокая степень допустимого параллелизма операций.
   4. Низкое энергопотребление схемы на одну операцию.
   5. Возможность перехода к жесткой логике на основе спроектированной схемы.
   6. Возможность работы с форматами чисел различной - произвольной точности.
   7. Высокая степень допустимого числа переиспользований данных на одну операцию чтения из памяти.
   8. **Простота работы с последовательными операциями.**
   9. **Возможность выпуска - приобретения устройств с более широким перечнем встроенных специализированных операций (оптимально реализованных в жёсткой логике).**
   10. **Более высокая скорость разработки программного обеспечения.**
3. Выберете устройства **исключительно** параллельной логики.
   1. **FPGA (Field-programmable gate array).**
   2. **CPLD (Complex programmable logic device).**
   3. SoC (System on chip).
   4. CPU (Central processor unit).
   5. DSP (Digital signal processor).
   6. GPGPU (General purpose graphical processor unit).
   7. **PLA (programmable logic array).**
   8. TPU (Tensor processor unit).
   9. NPU (Neural processor unit).
   10. **ASIC (application specific integrated circuit).**
4. Выберете утверждения, которые **не соответствуют** **унарной позиционной** системе счисления.
   1. Одна цифра (один знак) имеет разное значение в зависимости от места, где он поставлен.
   2. Десятичная система счисления является унарной позиционной.
   3. **Римская система счисления является унарной позиционной.**
   4. В позиционных система счисления числа всегда располагаются в таком порядке, что чем сильнее число, тем оно дальше от начала (или вправо или влево).
   5. **Система счисления черточками (чем больше черточек, тем больше число) является позиционной.**
   6. **Закодированные системы счисления всегда являются позиционными.**
   7. Системы счисления с плавающей запятой является унарной позиционной.
   8. **Унарные позиционные системы счисления допускают предоставление одной цифры в несколько позиций.**
   9. Унарные позиционные системы счисления не допускают предоставление одной цифры в несколько позиций.
   10. Представление числа в виде , где – цифра в системе счисления,   
       а – основание системы является формой унарного позиционного счисления.
5. Выберете утверждения, которые **не соответствуют** базовым логическим операциям.
   1. **Операция исключающее или – и является базовой логической операцией.**
   2. Операция и-не является базовой логической операцией
   3. Базис любой логической функции может быть задан либо как «и-не» либо как «или-не», то есть любую логическую операцию можно свести к одной из выше определенных, но не меньше.
   4. **Базис любой логической функции может быть задан либо как «и-не, исключающее или» либо как «или-не, исключающее или», то есть любую логическую операцию можно свести к одной из выше определенных, но не меньше.**
   5. Базовые логические схемы, соответствующие базовым логическим операциям, чаще всего реализованы на базе транзисторов в ключевом режиме.
   6. Базовые логические схемы, соответствующие базовым логическим операциям, представляют основу любого цифрового логического устройства, как жесткой, так и гибкой логики.
   7. Каждая логическая ячейка PLD ПЛИС основана на матрице базовых логических операций.
   8. **Арифметические операций над числами в позиционной унарной системе счисления требуют дополнить базовые логические операции элементом-сумматором.**
   9. **Элемент побитового сдвига не может быть реализован на базе логических операций и-не, или-не.**
   10. Элемент побитового сдвига может быть реализован на базе логических операций и-не, или-не.
6. Выберете утверждения, которые **не соответствуют** логическим триггерным устройствам.
   1. Триггеры являются устройствами, сохраняющими свое состояние выхода для заданного состояния входа.
   2. Триггеры чаще всего имеют два выхода.
   3. **Триггеры часто построены на базисе «исключающее или – не».**
   4. Триггеры могут иметь два основных входа, но могут иметь и два дополнительных RS входа.
   5. **Триггеры представляют собой параллельные логические элементы.**
   6. Триггеры представляют собой последовательные логические элементы.
   7. **T-триггеры могут не иметь синхросигнала.**
   8. D-Триггер представляет собой одно-битную ячейку памяти.
   9. Т-Триггер может иметь R и S входы.
   10. **T-триггеры не могут быть построены из других типов триггеров.**
7. Выберете утверждения, которые **не соответствуют** логическим устройствам на основе   
   D-Триггеров.
   1. Устройства на основе D-Триггеров могут быть как параллельными, так и последовательными.
   2. Регистры — это устройства хранения-преобразования входной информации, построенные на основе D-Триггеров.
   3. Ячейки памяти вида static RAM (SRAM) представляют собой параллельную сборку на D-Триггерах.
   4. устройства на основе D-Триггеров могут иметь R и S входы.
   5. **Шифраторы — это параллельные устройства на основе D-Триггеров.**
   6. **Демультиплексоры — это последовательные устройства на основе   
      RS-Триггеров.**
   7. **Регистры должны быть синхронными.**
   8. Каждый D-Триггер в регистре представляет собой один разряд.
   9. **На основе регистров не могут быть организованы устройства временной задержки импульсов.**
   10. На основе D-Триггеров могут быть организованы устройства временной задержки импульсов.
8. Выберете утверждения, которые **не соответствуют** логическим параллельным устройствам жесткой логики.
   1. Исключающее или является сумматором без переносов.
   2. Арифметико-логическое устройство основано на базе логических параллельных устройств.
   3. **Каждая логическая ячейка FPGA ПЛИС основана на логических параллельных устройства.**
   4. **Шифраторы используются для преобразования одного позиционного кода в другой.**
   5. Дешифраторы используются для преобразования не позиционного кода в позиционного.
   6. **Мультиплексор используются для преобразования позиционного кода в не позиционный.**
   7. Демультиплексоры переключают сигнал с одного информационного входа на информационный выход, указанный по входному адресу.
   8. Дешифраторы формируют «one-hot» выход.
   9. Мультиплексоры имеют один выход.
   10. **Мультиплексоры имеют «one-hot» информационные входы.**
9. Выберете утверждения, которые **не соответствуют** логическим параллельным устройствам гибкой логики.
   1. Программируемые логические интегральные схемы могут быть две программируемых матрицы «И» и «ИЛИ» либо только одну их них.
   2. **Программируемые логические интегральные схемы имеют только параллельные устройств жесткой логики в составе.**
   3. Программируемые логические интегральные схемы могут объединять в составе FPGA и CPLD блоки.
   4. **В программируемых логических интегральных схемах все логические блоки срабатывают одновременно.**
   5. **Программируемые логические интегральные схемы не позволяют организовать на своей основе последовательные устройства.**
   6. Макроячейки представляют собой блоки CPLD устройств, ориентированные на реализацию отдельных логических операций.
   7. В основе работы логических блоков FPGA лежит SRAM регистр.
   8. Устройства FPGA реализуются по принцу организации Look-up Table.
   9. **Устройства CPLD больше подходят для реализации сложных функций.**
   10. Устройства жесткой логики типа ASIC могут быть построены на основе как CPLD, так и FPGA ПЛИС.
10. Выберете утверждения, которые **не соответствуют** принципу программного управления устройством гибкой логики.
    1. **Принцип программного управления подходит как для последовательных, так и для параллельных устройств гибкой логики.**
    2. Принцип программного управления предполагает выполнение устройством (или одной его функциональной частью) по одной операции за один цикл работы.
    3. Для реализации принципа программного управления устройство должно иметь память для хранения входных инструкций и данных, а также для хранения выходных данных.
    4. **Принцип программного управления предполагает, что входные операции не могут быть выполнены параллельно в рамках каждой функциональной части.**
    5. **Принцип программного управления предполагает, что все операции должны быть выполнены в порядке, заданном программистом.**
    6. Принцип программного управления определяет каждую инструкцию, как конкатенацию кода операции, указания на операнды в заданном формате и служебную информацию.
    7. Инструкция программного принципа управления должна иметь аппаратную реализацию (в жесткой логике).
    8. **Устройства, представляющие конфигурируемый конечный автомат – являются устройствами с программным принципом управления.**
    9. Устройства, представляющие конфигурируемый конечный автомат – не являются устройствами с программным принципом управления.
    10. Машинный код представляет собой набор инструкций в заданной программистом-компилятором последовательности.
11. Выберете устройства, относящиеся к устройствам ввода-вывода ЭВМ.
    1. Центральный процессор.
    2. ОЗУ (оперативное запоминающее устройство, RAM).
    3. Чипсет системной платы.
    4. **Накопитель на жестких магнитных дисках (НЖМД).**
    5. **Клавиатура и мышь.**
    6. Основная память ЭВМ.
    7. Память программ, реализованная на ПЗУ (постоянном запоминающем   
       устройстве, ROM).
    8. Память данных.
    9. **Графический ускоритель.**
    10. **Сетевая карата.**
12. Выберете утверждения, которые **не соответствуют** принципу иерархии памяти.
    1. **Кэш память самый быстрый тип памяти.**
    2. Принцип минимума латентности лежит в основе иерархии памяти**.**
    3. **Иерархия памяти не связана с объемом памяти.**
    4. **Технология изготовления не связана с иерархией памяти.**
    5. Иерархия памяти связана с ценой на байт памяти.
    6. Внешняя память находится в конце иерархии памяти.
    7. SRAM быстрее DRAM.
    8. Регистровая память имеет самый маленький объем.
    9. **ОЗУ RAM самый дешевый тип памяти.**
    10. Кэш память построена по технологии SRAM.
13. Выберете **преимущества Гарвардской** архитектуры по сравнению с Фон-Неймана (Принстонской) архитектурой.
    1. Простота конвейеризации суб-оперпаций.
    2. Меньшая длина адреса позволяет сократить латентность.
    3. **Узким местом архитектуры является последовательный доступ к инструкциям и данных.**
    4. **Устройство процессора проще так как можно мультиплексировать шины данных и адреса**
    5. **Устройство процессора дешевле, за счет возможности использовать меньше контроллеров памяти и меньшего количества выводов для внешней памяти.**
    6. Инструкция может быть исполнена за один такт.
    7. Классическая реализация архитектуры требует как наличие ПЗУ, так и ОЗУ.
    8. **Возможность оперативного перераспределения памяти под нужды приложения.**
    9. Возможность исполнения одновременно нескольких инструкций.
    10. Особенности архитектуры объясняют ее использование в микроконтроллерных устройствах и устройствах быстрых вычислительных со-процессоров.
14. Выберете утверждения, которые **не соответствуют** особенностям КЭШ памяти.
    1. Возможны несколько уровней кэш памяти.
    2. Кэш память может делиться на память данных и память программ.
    3. Кэш-память чаще всего имеет ассоциативный или блочно-ассоциативный тип доступа.
    4. В кэше храниться последние и наиболее часто используемые операции.
    5. Чем выше уровень кэш-памяти, тем она большего объема.
    6. **Кэш-память выполнена по технологии dynamic-ram (DRAM).**
    7. **Кэш-память различных внутри-процессорных ядер не связана для типичных процессоров**.
    8. **Кэш-память на разных уровнях хранит разные данные.**
    9. **Кэш-память и ОЗУ имеют примерно равную латентность.**
    10. Кэш-память работает как асинхронный буфер по отношению к ОЗУ.
15. Выберете утверждения, которые **не соответствуют** особенностям набора регистров центрального процессора.
    1. Операции переходов соответствуют записи новых значений в счетчике команд.
    2. Состояние процессора определяется регистром флагов процессора.
    3. АЛУ выбирает операнды и помещает результаты в регистры общего назначения.
    4. **Регистровая память и кэш память имеют одинаковую латентность.**
    5. Наборы команд, поддерживаемые процессором непосредственно связаны с наборами регистров процессоров.
    6. **Все регистры, включенные в современные процессоры, необходимы в современных приложениях.**
    7. **Все регистры процессора программно доступны.**
    8. Состояние регистров для разных подпрограмм может быть разным.
    9. **Все регистры должны исполнять функции, для которых они предназначены.**
    10. Особенности регистров различаются для разных версий процессоров.
16. Выберете утверждения, которые **не соответствуют** структуре процессора Фон-Неймана.
    1. АЛУ включает операции побитового сдвига.
    2. Шины адреса и шины данных могут быть мультиплексированы.
    3. Все процессоры наследуют всю структуру предыдущих поколений в силу принципа обратной совместимости.
    4. УУ включает операции захвата-декодирования инструкций, выборки операндов и распределения операций.
    5. Процессор может иметь набор сопроцессоров, также управляемых УУ.
    6. Процессор имеет магистральную организацию.
    7. **Регистры для сопроцессоров и для основного процессора идентичны.**
    8. **УУ включает в себя регистры.**
    9. **АЛУ включает в себя регистры.**
    10. **В целях параллельности можно использовать несколько УУ.**
17. Выберете утверждения, которые **не соответствуют** типичным ISA (наборам команд) процессора.
    1. **CISC (полный) набор команд наиболее современный вариант ISA – по этому он используется в большинстве ЦПУ Intel и AMD.**
    2. **CISC (полный) набор команд имеет фиксированную длину команды и фиксированное время выполнения.**
    3. MISC (минимальный) набор команд усложняет компиляцию в машинный код.
    4. В современных CPU CISC команда разделяется на набор микроопераций формата RISC.
    5. **VLIW (длинное слово) набор команд выполняет несколько операций над одними данными.**
    6. Для CISC набора команд характерна работа только с операндами в регистрах.
    7. **Полный набор CISC команд требует больше регистров, чем RISC (сокращённый набор команд).**
    8. Особенность CISC команд – большинство из CISC команд могут быть заменены аналогами.
    9. MISC процессоры (с минимальным) набором команд упрощают процессор.
    10. Часть нагрузки УУ во VLIW процессорах перенесена в компилятор.
18. Выберете утверждения, которые **не соответствуют** типичным RISC ISA (сокращенный набор команд) процессора.
    1. Каждая RISC команда выполняется за один машинный цикл.
    2. Каждая RISC инструкция имеет фиксированную длину.
    3. RISC инструкции работают только с регистрами.
    4. **RISC инструкции включат большинство CISC (полный набор) инструкций, чтобы не терять эффективность процессора.**
    5. **Сокращенный набор команд RISC имеет и число регистров меньше чем CISC.**
    6. **Все регистры RISC имеют строгое назначение.**
    7. **RISC набор команд имеет более сложную структуру УУ чем CISC.**
    8. Основной недостаток RISC набора команд – необходимость формирования сложных команд компилятором.
    9. RISC инструкции ориентированы на выполнение команд конвейером.
    10. RISC набор команд наиболее современный.
19. Выберете утверждения, которые **не соответствуют микроархитектуре** суперскалярных конвейерныхпроцессоров**.**
    1. Незапланированные условные переходы одна из основных проблем конвейерного подхода.
    2. Исполнительные блоки суперскалярного конвейера оптимизированы под работу с RISC операциями.
    3. Конвейер может одновременно выполнять несколько разнотипных вычислительных операций.
    4. **Суперскалярный и конвейерный подход альтернативны.**
    5. **Суперскалярный конвейер лучше всего работает с VLIW инструкциями.**
    6. Суперскалярный конвейер защищается от проблем связанности данных в процессе работы.
    7. **Все декодированные команды суперскалярного конвейера выполняются одновременно.**
    8. Предсказание условных переходов выполняется чаще всего выполняется аппаратно, путем оценки вероятности перехода.
    9. Спекулятивное исполнение команд желательно не использовать в целях безопасности системы.
    10. **Внеочередное исполнение команд формируется при компиляции.**
20. Выберете утверждения, которые **не соответствуют** микроархитектуре конвейерных многопоточных процессоров.
    1. Многопоточность конвейера необходима для скрытия латентности таких операций как доступ к памяти.
    2. **Любой процессор с несколькими конвейерами** **может реализовать одновременную многопоточность.**
    3. Любой процессор с несколькими конвейерами может реализовать последовательную мелкозернистую многопоточность.
    4. **Сопроцессорные команды и команды основного datapath выполняются всегда в разных потоках.**
    5. **VLIW процессоры могут быть только многопоточными.**
    6. **Любая многопоточность безопасна.**
    7. Многоядерный процессор может быть рассмотрен как альтернатива многопоточному процессору.
    8. Распределение потоков операций по конвейерным модулям производится на аппаратном уровне.
    9. Распределение потоков производится на уровне микроопераций.
    10. Могут быть выделены временная мелкозернистая и крупнозернистая виды многопоточности.
21. Выберете утверждения, которые **не соответствуют** анализутаксономии Флина.
    1. Основной data path (АЛУ + регистры) архитектуры Фон-Неймана имеет SISD структуру.
    2. SIMD процессоры лучше всего подходят для обработки массивов данных.
    3. Многоядерные процессоры это MIMD процессоры.
    4. **Графические ускорители относятся к MIMD процессорам.**
    5. Современные процессорные ядра включают SIMD блоки.
    6. **Компьютерные кластеры не описываются классификацией Флина.**
    7. **SIMD архитектуры могут быть разделены на архитектуры с однородным, не однородным и только-кэш доступом к памяти.**
    8. MIMD архитектуры могут быть разделены на архитектуры с однородным, не однородным и только-кэш доступом к памяти.
    9. **Суперскалярная архитектура это MISD архитектура.**
    10. Векторные процессоры это MIMD процессоры.
22. Выберете утверждения, которые **не соответствуют** NUMA, UMA и COMA процессорам.
    1. Мультикомпьютерные системы не попадают под классификацию NUMA, UMA и COMA.
    2. Многоядерные системы часто представляют собой СС-NUMA системы.
    3. Кэш-камерность систем необходима для контроля связанности данных.
    4. **Большинство современных многоядерных серверных систем - COMA системы.**
    5. **Большинство современных бытовых процессоров – SMP-UMA** **системы.**
    6. Гетерогенные системы имеют единое адресное пространство независимо от того, какие модули включены в систему.
    7. **Гетерогенные HSA системы имеют одинакове уровни Кэш памяти для разных модулей.**
    8. **Big.Little гетерогенность предполагает наличие разных наборов команд для разных модулей.**
    9. Гетерогенные HSA системы предполагают наличие разных ISA для разных модулей.
    10. В кэш-когерентных системах все устройства обмениваются системными сообщениями по общему интерфейсу.
23. Выберете утверждения, которые **не соответствуют** тенденциям современных центральных процессорных устройств.
    1. **Увеличение тактовой частоты ядра.**
    2. **Увеличение напряжения на ядре за счет более современных технологий теплоотвода.**
    3. **Отказ от многопоточности.**
    4. **Отказ от суперскаляризации в силу небезопасности.**
    5. Переход к динамическому предсказанию ветвлений от спекулятивного исполнения инструкций.
    6. Использование гетерогенных СС-NUMA архитектур.
    7. Использование многоуровневых систем доступа к памяти.
    8. Расширение набора сопросцессоров работы с массивами данных.
    9. Расширение числа функциональных модулей микроархитектуры.
    10. Расширение числа ступеней конвейера.
24. Выберете утверждения, которые **не соответствуют** тенденциям современных графический сопроцессоров (GPU).
    1. Для интегрированных GPU предпочтительно использовать тоже адресное празднество, что и для CPU.
    2. Основные операции, выполняемые на GPU – это сложнее с умножением для наборов чисел с плавающей запятой.
    3. Современные GPU представляют собой процессор типа один поток инструкций, много данных.
    4. Основное узкое место GPU – латентность доступа к ОЗУ.
    5. Тензорные блоки GPU позволяют обрабатывать данные в форматах смешанной точности.
    6. **Все связки потоков GPU могут быть запущены одновременно или по раздельности.**
    7. **Тензорные блоки GPU ориентированы на оптимизацию расчета текстур.**
    8. **Современные** **GPU представляет собой единую структуру – массив потоковых процессов.**
    9. Современные GPU представляют собой набор мультипроцессоров, работающих независимо.
    10. **Каждый потоковый процессор GPU может быть запущен независимо от других.**
25. Выберете утверждения, которые **не соответствуют** тенденциям современных процессоров специального назначения.
    1. **Современные микроконтроллеры ориентированы на многозадачную работу.**
    2. Для микроконтроллеров не требуется использование операционной системы.
    3. Современные микроконтроллеры включают широкий набор сопроцессоров и модулей работы с устройствами ввода вывода и памятью.
    4. Квантовые процессоры позволяют повысить производительность в операциях криптографии при возможности параллелизации алгоритмов.
    5. Нейроморфные процессоры включают большое число простых упорядоченных выполняющих блоков процессоров.
    6. Тензорные процессоры имеют архитектуру типа систолический массив.
    7. **Современные микроконтроллеры имеют фон-неймановскую архитектуру.**
    8. **Современные процессоры мобильных устройств поддерживают CISC операции для совместимости с ЦПУ общего назначения.**
    9. **Современные процессоры мобильных устройств имеют суперскалярную конвейерную многопоточную микроархитектуру.**
    10. Тензорные процессоры ориентированы на максимизацию переиспользования данных.