1. (???)
2. Uma arquitetura do tipo Havard é caracterizada por permitir o acesso a instruções e dados do mesmo ciclo de relógio.
3. Um endereço de memória externa num sistema computacional é um número único que identifica cada posição de memória.
4. Espaço de endereçamento de memória num sistema computacional é a gama total de posições de memória que o CPU pode referenciar.
5. Numa memória com uma organização do tipo byte-addressable: A cada endereço está associado um dispositivo de armazenamento de 1 byte.
6. A arquitetura MIPS é caracterizada por ser do tipo load-store
7. A arquitetura MIPS é caracterizada por possuir 32 registos de uso geral de 32 bit cada, ser do tipo load-store e possuir poucos formatos de instrução.
8. A arquitetura MIPS é do tipo “Load-Store”. Isso significa que: Os operandos das operações aritméticas e lógicas apenas podem residir em registos internos.
9. Na arquitetura MIPS, os campos de uma instrução do tipo “R” são do tipo:

* Op,rs,rt,rd,shamt,funct

1. Na arquitetura MIPS, os campos de uma instrução do tipo ”I” são do tipo:

* Op,rs,rt,endereço (offset/imm)

1. Nas instruções de acesso à memória da arquitetura de MIPS é utilizado o modo de endereçamento indireto por registo.
2. No MIPS, a instrução de salto incondicional indireto através de registoé codificada usando o formato de codificação J.
3. Quando um endereço se obtém da adição do conteúdo de um registo com um offset constante diz se que estamos perante um endereçamento indireto a registo com deslocamento.
4. O formato de instruções do tipo “I” da arquitetura MIPS é usado nas instruções de, salto condicional, aritméticas em que somente um dos operandos está armazenado num registo, acesso à memória de dados externa, etc..
5. Na instrução lb da arquitetura MIPS, o operando é obtido através de endereçamento indireto a registo com deslocamento.
6. Nas instruções do tipo R da arquitetura MIPS É utilizado o modo de endereçamento registo.
7. O modo de endereçamento utilizado na instrução sb $8,-8($s8) é indireto por registo com deslocamento.
8. A instrução virtual “li $t0, 0x100012345” da arquitetura MIPS decompõe-se na seguinte sequência de instruções nativas: “lui $t1,0x1001” seguida de “ori $t0,$t1,0x2345”
9. A instrução virtual “bgt $t8,$t9,target” da arquitetura MIPS decompõe-se na seguinte sequência de instruções nativas: d“slt $1,$t9,$t8” seguida de “bne $1,$0,target.