# Universidad Tecnológica de Pereira Programa de Ingeniería de Sistemas y Computación

**Asignatura IS614 Arquitectura de Computadores Taller de Quices – 27 Mayo de 2020**

1. Un algoritmo toma 10 segundos en ser ejecutado en un Computador A, el cual tiene una frecuencia de reloj de 2 GHz. Se quiere diseñar un computador B que ejecute el mismo algoritmo en 6 segundos. El diseñador ha determinado que se puede hacer una mejora sustancial en la frecuencia del reloj, pero haciendo que la cantidad promedio de ciclos por instrucción sea aumentado en un factor de 1.2 comparado con el Computador A. ¿Cuál debe ser la frecuencia de reloj que debe tener el Computador B para cumplir el objetivo de diseño? (25%)
2. Convierta la siguiente instrucción de arquitectura ARMv8 que se encuentran en lenguaje máquina en formato hexadecimal a una sentencia correspondiente en lenguaje de alto nivel C. (25%)

0xCB1702D5

1. Basándose en un procesador monociclo de arquitectura ARMv8 como el visto en clase explique en funcionamiento de cada una de las señales de control y construya una tabla con valores que toman las señales de control para cada una de las instrucciones compatibles. (25%)
2. Convierta el número 328.625 a su representación en punto flotante en precisión simple IEEE754. (25%)

NIT: 891.480.035-9 / Código Postal: 660003 / Conmutador: +57(6)3137300 / Fax: 3213206 / Dirección: Cra 27 Nº 10-02, Los Álamos, Pereira, Risaralda, Colombia / [www.utp.edu.co](http://www.utp.edu.co/) Reacreditada como Institución de Alta Calidad por el Ministerio de Educación Nacional 2013-2017

1. Un algoritmo toma 10 segundos en ser ejecutado en un Computador A, el cual tiene una frecuencia de reloj de 2 GHz. Se quiere diseñar un computador B que ejecute el mismo algoritmo en 6 segundos. El diseñador ha determinado que se puede hacer una mejora sustancial en la frecuencia del reloj, pero haciendo que la cantidad promedio de ciclos por instrucción sea aumentado en un factor de 1.2 comparado con el Computador A. ¿Cuál debe ser la frecuencia de reloj que debe tener el Computador B para cumplir el objetivo de diseño? (25%)

Tiempo de ejecución A = 10s

Tiempo de ejecución B = 6s

Frecuencia A = 2GHz

Diferencia ciclos por instrucción = 1.2

Frecuencia B = 4GHz

1. Convierta la siguiente instrucción de arquitectura ARMv8 que se encuentran en lenguaje máquina en formato hexadecimal a una sentencia correspondiente en lenguaje de alto nivel C. (25%)

0xCB1702D5

1100 1011 0001 0111 0000 0010 1101 0101

C B 1 7 0 2 D 5

11001011000 -10111-000000-10110-10101

SUB X23 0 X22 X21

Long long int X = long long int Y – long long int Z;

X 🡪 X23

Y 🡪 X22

Z 🡪 X21

1. Basándose en un procesador monociclo de arquitectura ARMv8 como el visto en clase explique en funcionamiento de cada una de las señales de control y construya una tabla con valores que toman las señales de control para cada una de las instrucciones compatibles. (25%)

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **SEÑALES DE CONTROL** | | | | | | | | | | |
|  |  | **PCSrc** | **DMWr** | **DMRd** | **RFDataWrScr** | **ALUOp** | **ALUBScr** | **SEUScr** | **RFWr** | **RegWrScr** | **FlagsWr** | **RegRd2Scr** |
| **INSTRUCCIONES** | **ADD** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| **SUB** | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| **AND** | 0 | 0 | 0 | 0 | 10 | 0 | 0 | 1 | 0 | 0 | 0 |
| **OR** | 0 | 0 | 0 | 0 | 11 | 0 | 0 | 1 | 0 | 0 | 0 |
| **EOR** | 0 | 0 | 0 | 0 | 100 | 0 | 0 | 1 | 0 | 0 | 0 |
| **NOR** | 0 | 0 | 0 | 0 | 101 | 0 | 0 | 1 | 0 | 0 | 0 |
| **LSL** | 0 | 0 | 0 | 0 | 110 | 0 | 0 | 1 | 0 | 0 | 0 |
| **LSR** | 0 | 0 | 0 | 0 | 111 | 0 | 0 | 1 | 0 | 0 | 0 |
| **ADDI** | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| **SUBI** | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| **ANDI** | 0 | 0 | 0 | 0 | 10 | 1 | 0 | 1 | 0 | 0 | 0 |
| **ORI** | 0 | 0 | 0 | 0 | 11 | 1 | 0 | 1 | 0 | 0 | 0 |
| **EORI** | 0 | 0 | 0 | 0 | 100 | 1 | 0 | 1 | 0 | 0 | 0 |
| **LSUR** | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 |
| **STUR** | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| **B** | 1 | 0 | 0 | 10 | 1000 | 0 | 10 | 0 | 0 | 0 | 0 |
| **BL** | 1 | 0 | 0 | 10 | 1000 | 0 | 10 | 1 | 1 | 0 | 0 |
| **CBZ** | FlagsZ | 0 | 0 | 10 | 1000 | 1 | 11 | 0 | 0 | 0 | 1 |
| **CBNZ** | ~FlagsZ | 0 | 0 | 10 | 1000 | 1 | 11 | 0 | 0 | 0 | 1 |
| **B.cond** | Cond&FlagsR | 0 | 0 | 10 | 1000 | 0 | 10 | 0 | 0 | 0 | 0 |
| **ADDS** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| **SUBS** | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| **ANDS** | 0 | 0 | 0 | 0 | 10 | 0 | 0 | 1 | 0 | 1 | 0 |
| **ADDIS** | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| **SUBIS** | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 |
| **ANDIS** | 0 | 0 | 0 | 0 | 10 | 1 | 0 | 1 | 0 | 1 | 0 |

PCScr: esta señal de control toma valores dependiendo del tipo de instrucción para determinar cual va a ser el siguiente valor del program counter. Es 0 para instrucciones tipo R, D e I, es 1 para tipo B y depende de las señales de Cond y Flags para determinar el valor en el tipo CB.

DMWr: esta señal de control toma valores dependiendo de si se quiere escribir o no en la data memory, 1 si se escribe, 0 si no.

DMRd: esta señal de control toma valores dependiendo de si se quiere leer o no de la data memory, 1 si se lee, 0 si no.

RFDataWrSrc: esta señal de control es la entrada para el multiplexor que decide cual va a ser el valor del DataWr del Register File, si es 0 toma la salida de la ALU, si es 1 toma la salida de la data memory, y si es 2 toma la dirección siguiente del program counter

ALUOp: esta señal de control de 4 bits toma valores según la operación artimetica o lógica que se vaya a realizar, o si se va a realizar un salto.

ALUBScr: esta señal de control toma valores según la instrucción que llegue a la control unit, si es 0 es porque es una instrucción que opera entre registros, si es 1 es porque opera entre un registro y la salida de la SEU.

SEUScr: esta señal de control toma valor dependiendo del tipo de instrucción y lo que hace es extender determinados bits de la entrada de la SEU hasta completar 64.

RFWr: esta señal de control es la que indica al register file si efectivamente se quiere escribir en el register file si la instrucción así lo requiere, es 1 si se escribe, 0 si no.

RegWrScr: esta señal de control es el selector que escoge entre el registro Rt y el X30, se usa cuando se tienen instrucciones de tipo Branch and Link (BL).

FlagsWr: esta señal de control es la entrada habilitante del registro de las Flags, se usa cuando la instrucción tiene la regla de aplicar Flags.

RegRd2Scr: esta señal de control es el selector del multiplexor que escoge entre Rm y Rt para ser la entrada RegRd2 del register file, esto depende del tipo de instrucción.

1. Convierta el número 328.625 a su representación en punto flotante en precisión simple IEEE754. (25%)

Se pasa a Binario 328

328.625 = 101001000

Se transforma el numero binario en formato científico

101001000 = 1,0100100 x 28

Se halla el exponente.

Exponente: 8+127 = 135 🡪 a binario: 10000111

Se halla la mantisa hasta llegar a 23 bits

Mantisa: 1,0100100= 01001000000000000000000

Se halla el signo.

Signo=0

328.625 en IEEE754 Signo (0)-Exponente (10000111)-Mantisa (01001000000000000000000)

01000011101001000000000000000000