

Instituto Tecnológico y de Estudios Superiores de Monterrey



**Tecnológico
de Monterrey**

Situación problema

Fecha: 20 de octubre del 2022

Modelación de sistemas mínimos y arquitecturas computacionales (Gpo 200)

Alumnos:

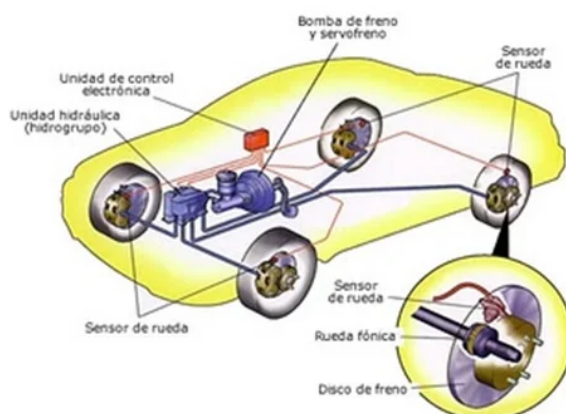
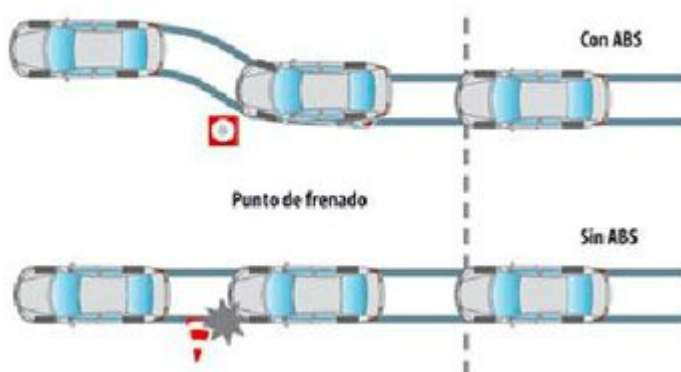
A01423897 Patricio Alvarez Hernandez - Patricio Álvarez Hernández

A01423940 Daniel De Luna Díaz - Daniel De Luna Díaz

Situación problema	0
Modelación de sistemas mínimos y arquitecturas computacionales (Gpo 200)	0
Antecedentes	2
Objetivo	3
Desarrollo	3
Propuesta	3
Agenda	3
Simulación/prototipo	4
Programa en ensamblador	4
Diagrama de flujo:	6
Código:	7
Documentación:	8
Usos:	8
Conclusiones	8
Referencias	9

Antecedentes

El frenado ABS, el cual es una función para poder maniobrar el vehículo mientras se está frenando para poder evitar obstáculos sin perder el control de este, este se hace gracias a que evita que las ruedas se desbloqueen y lleva a un frenado seguro, y en esta función hay varios problemas que se deben resolver pero uno de ellos son los fallos en las electroválvulas, las cuales son las encargadas de regular la presión del frenado, o también las que son por fallos en los sensores de la luz de frenado. En estos dos casos se puede analizar en una computadora, la de electroválvulas puede ser con un escaneo usando, aparte de la computadora, unos detectores para ver si el error viene de las electroválvulas o de otro sitio, y el de los sensores son más sencillos, es conectarlo a una computadora especial y escanear el funcionamiento correcto de estos.



Esquema de instalación del sistema A.B.S. en el vehículo

Objetivo

Nuestro objetivo es simular cómo detectaría el vehículo cuando se activen los frenos ABS, y usar el promedio para calcular qué tanta necesidad de los frenos se usará dependiendo de la velocidad del vehículo, para eso es usado el promedio.

Desarrollo

Propuesta

Para nuestra solución, proponemos lo siguiente tomando como referencia un sensor que entrega una lectura a través de un bus paralelo a razón de 12 bits/ms.

Desarrollar un programa que entregue la media de 10 datos, de alguno de los sensores en el puerto de salida de assembly MARIE el cual simule los sensores mencionados anteriormente con la finalidad de poder medir si estos siguen funcionando de manera adecuada o si deben de ser reemplazados.

Agenda

La primera semana nos centramos en investigar y aprender los conceptos básicos y principales de la computadora, viendo primero lo que vimos en clase, que abarca la investigación de la historia de las computadoras desde las tarjetas perforadas, pasando por las primeras "Personal Computers" (P.C.) hasta la tecnología que hay ahora, tales como, las V.R. y el mundo tan conectado en el que vivimos.

En la segunda semana ya empezamos con la investigación de la situación problema, empezamos investigando los principales problemas que se encuentran en la industria automotriz a través de la electrónica del vehículo, en los cuales investigamos varios temas, encontramos problemas con las bolsas de aire, malfuncionamientos de la batería, y el tema que mas centramos y que más llamó nuestra atención e interés es el de los frenos ABS, en el cual vimos que son muy importantes poder maniobrar mientras se están los frenos, en caso de fallar podría

pasar un accidente, por lo que fue un tema que nos parecía muy importante resolver.

La tercera semana empezamos a ver todo de cómo funciona la CPU y empezamos a ver el lenguaje de programación de ensamblador, por lo que esta semana no hubo mucho avance a la situación problema, pero nos empezamos a dar una idea de cómo podríamos planear el programa que se presentaría en esta entrega.

La cuarta semana seguimos viendo ensamblador y estábamos viendo lo de memoria, y en la situación problema empezamos a programar el código según lo visto en clase.

La quinta semana ya era terminar el programa en Marie y empezar a poder todo el reporte en orden y terminar la reflexión para poder presentar en condiciones todo nuestro proyecto.

Simulación/prototipo

Programa en ensamblador

Nuestra primera simulación fue con los siguientes datos:

1, 2, 3, 4, 5, 6, 7, 8, 9, 10.

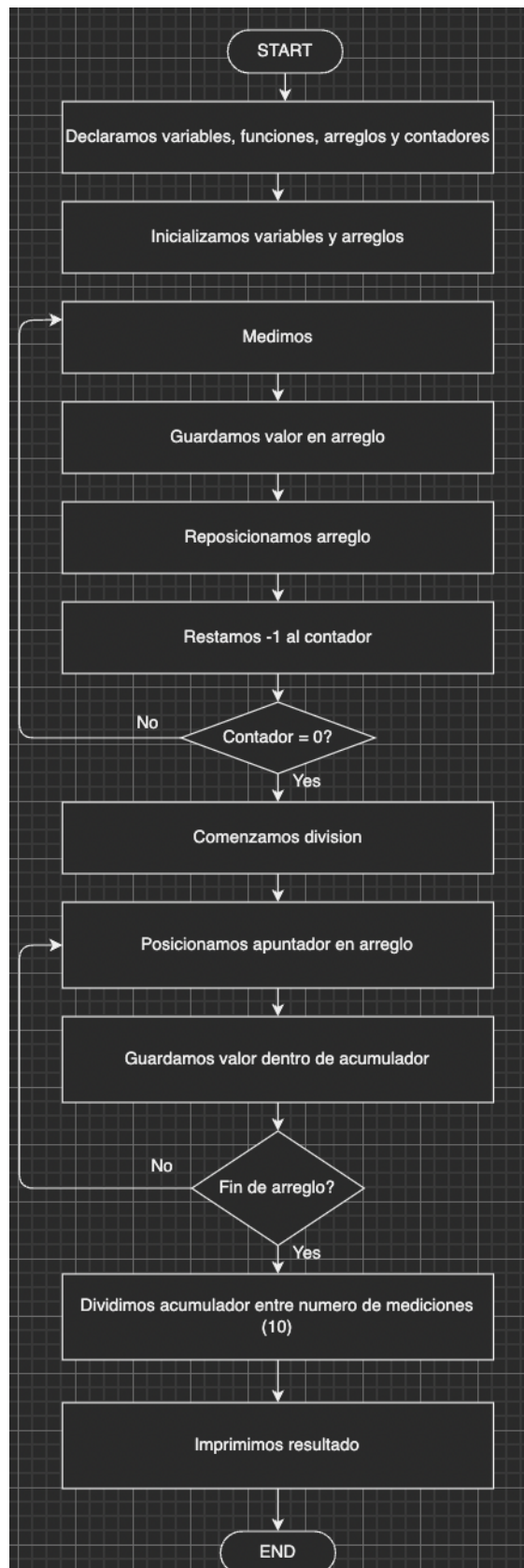
El cual nos arrojó un resultado de 6 es decir, adecuado.

The screenshot shows the Marie CPU simulator interface. The main window displays assembly code with comments in Spanish. The code includes instructions like `LOAD CONT_B`, `SUBT ONE`, `STORE CONT_B`, `SKIPCOND 400`, `JUMP SAVE`, `JUMP DIV`, `END`, `LOAD ANS`, `OUTPUT`, `HALT`, `VZERO`, `JUMP END`, `LOAD ANS`, `ADD ONE`, and `STORE ANS`. The right panel shows the state of the CPU registers: `AC` (0006), `IR` (7000), `MAR` (017), `MBR` (7000), `PC` (018), `IN` (0010), and `OUT` (0006). The bottom panel shows the memory dump, which is mostly zeroed out, with some non-zero values in the first few addresses.

```
Assembly code:
21 LOAD CONT_B /Cargamos el valor de CONT B
22 SUBT ONE /Le restamos -1 al contador B
23 STORE CONT_B /Almacenamos el valor del nuevo contador
24 SKIPCOND 400 /Si el contador B = 0 entonces ya terminamos
25 JUMP SAVE /Si el contador B /= 0 entonces seguimos
26 JUMP DIV /Si ya terminamos entonces comenzamos division
27
28 END, LOAD ANS /Cargamos el promedio en AC
29 OUTPUT /Imprimimos el promedio en AC
30 HALT /Terminamos el programa
31
32 /Funciones
33 /Verificamos si AC es igual a cero
34 VZERO, SKIPCOND 400
35 JUMP END /Si no es igual nos saltamos al final del programa
36 LOAD ANS /Si si es igual a 0, cargamos respuesta en AC
37 ADD ONE /Incrementamos AC + 1
38 STORE ANS /Guardamos nuevo valor en respuesta

Machine halted normally.
+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F
000 A000 1027 2028 102B 202B 202C 102A 2029 5000 E028 302B 202B 1029 302E 2029 1028
010 402E 2029 8400 9008 901E 102C 6000 7000 8400 9015 102E 302E 202C 901E 102B 4027
020 8000 9018 202B 102C 302E 202C 901E 000A 0000 005A 0050 0001 0006 0000 0001 0000
030 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
040 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
050 0001 0002 0003 0004 0005 0006 0007 0008 0009 0010 0000 0000 0000 0000 0000
060 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
070 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
080 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
090 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0A0 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0B0 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
```

Diagrama de flujo:



Código

/ Evidencia , Miercoles 19 de octubre, 2022
 / TC1032.200

CLEAR

```

START,  LOAD CONT_A    /Cargamos el contador A (10)
        STORE CONT_B  /Asignamos el valor de 10 a nuestro contador B
        LOAD ZERO      /Cargamos consante 0
        STORE ACU      /Asignamos el valor de 0 al acumulador
        STORE ANS      /Asignamos el valor de 0 a nuestra respuesta
        LOAD BCKP      /Cargamos BCKP que apunta al inicio de ARR
        STORE ARR      /Asignamos el valor a arr para sobre-escribir los nuevos datos

SAVE,   INPUT          /Leemos valor del teclado
        STOREI ARR     /Almacenamos valor leído en donde apunta ARR
        ADD ACU        /Sumamos el acumulador
        STORE ACU      /Almacenamos nueva suma en el acumulador
        LOAD ARR       /Cargamos posicion actual de ARR
        ADD ONE        /Le incrementamos +1 a la posicion del arreglo
        STORE ARR      /Guardamos la nueva posicion del arreglo
        LOAD CONT_B    /Cargamos el valor de CONT B
SUBT ONE /Le restamos -1 al contador B
        STORE CONT_B  /Almacenamos el valor del nuevo contador
        SKIPCOND 400   /Si el contador B = 0 entonces ya terminamos
        JUMP SAVE     /Si el contador B /= 0 entonces seguimos
        JUMP DIV      /Si ya terminamos entonces comenzamos division

END,    LOAD ANS       /Cargamos el promedio en AC
        SUBT EXP       /Restamos promedio con expectativa
        SKIPCOND 400   /Si < o > a 0 entonces imprimimos error
        JUMP PRINT     /Imprimimos error
        SKIPCOND 800   /Si > a 0 entonces imprimimos error
        JUMP END2      /Imprimimos verde
        JUMP PRINT     /Imprimimos error

END2,   LOAD MEN2     /Cargamos color verde
        OUTPUT        /Imprimimos todo bien
        JUMP START    /Reiniciamos programa
    
```

/Funciones

/Verificamos si AC es igual a cero

```

VZERO,  SKIPCOND 400
        JUMP END      /Si no es igual nos saltamos al final del programa
        LOAD ANS      /Si si es igual a 0, cargamos respuesta en AC
        ADD ONE       /Incrementamos AC + 1
        STORE ANS     /Guardamos nuevo valor en respuesta
        JUMP END      /Nos vamos al final del programa
    
```

/Dividimos para sacar el promedio

```

DIV,    LOAD ACU      /Cargamos el valor del acumulador en AC
        SUBT CONT_A   /Le restamos CONT A a AC
        SKIPCOND 800  /Si el numero es mayor a 0 seguimos
        JUMP VZERO    /Si no es mayor a cero nos saltamos a la funcion VZERO
        STORE ACU     /Almacenamos nuevo valor en el acumulador
        LOAD ANS      /Cargamos el valor de ANS en AC
        ADD ONE       /Incrementamos el valor de AC + 1
        STORE ANS     /Almacenamos el valor de AC en ANS
        JUMP DIV      /Iteramos para seguir con el ciclo
    
```

```

PRINT,  LOAD MEN1    /Cargamos color amarillo
        OUTPUT      /imprimimos error amarillo
    
```

JUMP START	/Reiniciamos programa
/Variables	
CONT_A, DEC 3	/Contador de 10 medidas
CONT_B, DEC 0	/Contador generico
ARR, HEX 50	/Arreglo de valores medidos
BCKP, HEX 50	/Direccion de arreglo
ACU, DEC 0	/Acumulador
ANS, DEC 0	/Respuesta
/Variables constantes	
ZERO, DEC 0	/Constante = 0
ONE, DEC 1	/Constante = 1
EXP, DEC 5	
/Msgs	
MEN1, HEX 59	/ "Y" de yellow es decir error
MEN2, HEX 47	/ "G" de green es decir bien

Documentación

Para nuestra simulación se utilizaron un total de 5 ciclos y 8 variables, dos las cuales eran constantes.

Para nuestros ciclos se utilizaron 3 principales y 2 funciones;

- START;

Aquí es donde comenzamos el programa y donde leemos las medidas del sensor

- SAVE;

Aquí es donde guardamos las medidas leídas y posicionamos nuestro apuntador para guardar la siguiente medida

- END;

Cargamos nuestra respuesta y la imprimimos en pantalla para luego reiniciar el proceso

- VZERO;

Esta función simplemente verifica si nuestro valor en AC es 0 para saber si estamos cargando y guardando nuestras medidas de manera adecuada.

- DIV.

Esta función se encarga de recorrer el arreglo, sumar todos nuestros valores y obtener la media para guardarla en nuestra variable de respuesta.

Usos:

La ventaja de nuestra simulación es que a pesar de que esta se encuentra simulando las electroválvulas de nuestro vehículo, este puede ser aplicado en otras áreas de un coche desde medición de fluidos (Gas, aceite, anticongelante, etc.) o incluso los niveles de presión de las llantas, la calidad de un filtro de aire acondicionado y la esperanza de vida de la batería entre muchas otras.

Conclusiones

La industria automotriz ha tenido un avance muy grande desde que se empezó hasta ahora, todo con los primeros automóviles que eran a base de vapor y después de los primeros accidentes automovilísticos se inventaron los primeros freno de mano, las diferentes velocidades y el volante, y también por un calentamiento que se produce en la caldera se debió sustituir el vapor primero por aceite, por gas hasta que, llegó el primer vehículo a base de gasolina, en 1885 construido por Carl Benz, luego que crearía Mercedes-Benz, el Benz Patent-Motorwagen, y en en los 1800 era se empezaron a usar los primeros coches eléctricos, se basaba en el electromagnetismo de Faraday, sus ventajas eran que eran más ligeros y más silenciosos, pero tenía una principal desventaja que era su corta autonomía, y esa desventaja es por la cual los coches de combustión fueron mejor aceptados

Referencias

- López, N. (2015, March 14). Siete razones por las que puede fallar el ABS. Autobild.es. Retrieved September 25, 2022, from <https://www.autobild.es/contenido-patrocinado/especial-toyota-siete-razones-por-las-que-puede-fallar-abs-247839>
- N.A. (2021, August 29). ¿Qué significa frenos abs y cómo funcionan? - geely Costa Rica. Geely. Retrieved September 25, 2022, from <https://www.geely.cr/2020/03/10/blog-geely-varios-que-significa-frenos-abs-y-como-funcionan/>
- N. N. (2018, January 29). Hace 132 años nació el primer automóvil. Oficial Mercedes Benz. Retrieved October 19, 2022, from <https://lamercedpilar.com/universo-mb/hace-132-anos-nacia-el-primer-automovil#:~:text=El%2029%20de%20enero%20de,carruaje%20motorizado%20de%20cuatro%20ruedas>
- C. (2019, June 13). La Evolución del Automóvil. Conexión Industriales. Retrieved October 19, 2022, from <https://conexionindustriales.com/la-evolucion-del-automovil/>