Recuerdos del Instituto de Cálculo, UBA circa 1966

Cristina Zoltan

Depatament de Llenguatges i Sistemes Informàtics
Universitat Politècnica de Catalunya
Barcelona, España

13 Mayo 2011

Esquema de la presentación

Esquema de la presentación

Introducción

En esta charla presentaré algunos aspectos no demasiado difundidos del lenguaje desarrollado en el IC para la Mercury: el lenguaje Comic. Un primer aspecto corresponde a la presencia de una salida adicional a la máquina: salida gráfica.
A continuación me referiré a los mecanismos de reingeniería que disponíamos en la época.
Completaré la charla con una anécdota sobre la programación de un modelo ganadero, bajo la dirección de Oscar Varsavsky. Era mi primer gran programa en Comic.

El texto recopila la presentación

Esquema de la presentación

El graficado

- La entrada y la salida eran muy primitivas: cinta perforada de papel, impresora de línea ...

El graficado

- La entrada y la salida eran muy primitivas: cinta perforada de papel, impresora de línea ...
- El graficador

El graficado

- La entrada y la salida eran muy primitivas: cinta perforada de papel, impresora de línea ...
- El graficador
- Cabeza graficadora que giraba, se desplazaba dibujando o no y en tinta negra o un caracter de un conjunto de 16.

El graficado

- La entrada y la salida eran muy primitivas: cinta perforada de papel, impresora de línea ...
- El graficador
- Cabeza graficadora que giraba, se desplazaba dibujando o no y en tinta negra o un caracter de un conjunto de 16.
- Se agregó al COMIC las operaciones usuales de graficación: coordenadas, escalas, graficación de varias curvas

El graficado

- La entrada y la salida eran muy primitivas: cinta perforada de papel, impresora de línea ...
- El graficador
- Cabeza graficadora que giraba, se desplazaba dibujando o no y en tinta negra o un caracter de un conjunto de 16.
- Se agregó al COMIC las operaciones usuales de graficación: coordenadas, escalas, graficación de varias curvas
- Podía hacer solamente movimientos horizontales y verticales

El graficado

- La entrada y la salida eran muy primitivas: cinta perforada de papel, impresora de línea ...
- El graficador
- Cabeza graficadora que giraba, se desplazaba dibujando o no y en tinta negra o un caracter de un conjunto de 16.
- Se agregó al COMIC las operaciones usuales de graficación: coordenadas, escalas, graficación de varias curvas
- Podía hacer solamente movimientos horizontales y verticales
- Hubo que inventar para que las rectas no pareciera la gráfica de una función escalera

El graficador

La entrada y salida original de la Mercury era cinta de papel. A lo largo de los años se fueron agregando periféricos como la impresora de línea. El último en llegar fue el graficador.
El graficador era un dispositivo poco conocido para la época. Su uso implicó desarrollar funciones para que los usuarios pudieran utilizarlo, logrando definir un conjunto de funciones que permitiesen una sencilla utilización del graficador.
Se integró este dispositivo de salida a Comic y siendo éste un lenguaje de alto nivel, se desarrollaron funciones de alto nivel para el uso.
El graficador obedecía un conjunto de comandos de bajo nivel: moverse sin dibujar desde la posición actual a la posición indicada, hacer lo mismo pero dibujando, rotar la cabeza delineadora para elegir la pluma, etc.
Usando estos comandos básicos se diseñaron funciones para Comic (la familia Ψ GRAFICAR).

El texto recopila la presentación

El graficador

Una función era la de describir el dibujado de los ejes de coordenados, dando las escalas en las que se deseaban los (2) ejes. En ese momento solo se pensaba en gráficos en 2 dimensiones.
Graficar una función era una sola instrucción, a partir de una rutina que la calculaba. Opcionalmente los valores obtenidos se trataba como una colección de puntos en el plano, o como conjunto de puntos conectados.
Los movimientos producían líneas rectas, por lo que se desarrollaron algoritmos para lograr que las funciones "continuas" no lucieran como polígonos.
.

El texto recopila la presentación

El graficador

No creo que se usara demasiado este periférico ya que el desarrollo de estas funcionalidades y su integración en Comic constituyó mi trabajo del seminario superior (trabajo de fin de carrera) que presenté en los primeros días de mayo de 1966.....

El texto recopila la presentación

Esquema de la presentación

La memoria de Clementina

- La memoria operativa era de 1.024 palabras de 40 bits

La memoria de Clementina

- La memoria operativa era de 1.024 palabras de 40 bits
- Esta memoria tiene capacidad para 1.024 instrucciones (la memoria direccionable)

La memoria de Clementina

- La memoria operativa era de 1.024 palabras de 40 bits
- Esta memoria tiene capacidad para 1.024 instrucciones (la memoria direccionable)
- 2.048 números enteros en el rango (-512, 511). Se la considera formalmente dividida en 32 páginas, conteniendo cada página 32, 64 ó 128 palabras según se consideren éstas como de 40, 20 ó 10 bits.

La memoria de Clementina

- La memoria operativa era de 1.024 palabras de 40 bits
- Esta memoria tiene capacidad para 1.024 instrucciones (la memoria direccionable)
- 2.048 números enteros en el rango (-512, 511). Se la considera formalmente dividida en 32 páginas, conteniendo cada página 32, 64 ó 128 palabras según se consideren éstas como de 40, 20 ó 10 bits.
- Los números pueden estar representados en punto flotante dedicando 30 bits para la mantisa y 10 para el exponente, o como enteros cortos de 10 bits.

La memoria de Clementina

- La memoria operativa era de 1.024 palabras de 40 bits
- Esta memoria tiene capacidad para 1.024 instrucciones (la memoria direccionable)
- 2.048 números enteros en el rango (-512, 511). Se la considera formalmente dividida en 32 páginas, conteniendo cada página 32, 64 ó 128 palabras según se consideren éstas como de 40, 20 ó 10 bits.
- Los números pueden estar representados en punto flotante dedicando 30 bits para la mantisa y 10 para el exponente, o como enteros cortos de 10 bits.
- Las instrucciones son unidades de información de 20 bits

Inferencia de tipos Los post-mortem

La documentación sobre la Mercury, en lo que respecta a la programación, era manuales de usuario. No se tenía ninguna documentación de los código fuente de los ensambladores y el compilador con el cual se trabajaba.
Solo se tenía el código binario de ese software.
Pero una sucesión de bits tiene diferentes interpretaciones: puede ser una constante, una variable. En ese caso importa saber el tipo. Pero también puede ser una instrucción.
Veremos, sin entrar en detalles, la estructura de cada uno de los elementos que podíamos encontrar en esas secuencias de bits.

El texto recopila la presentación

Los enteros

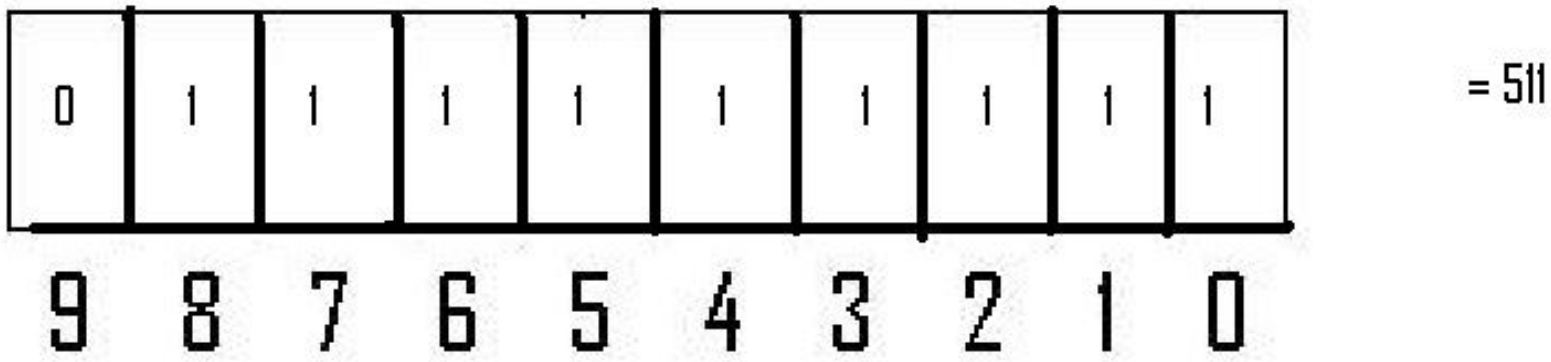
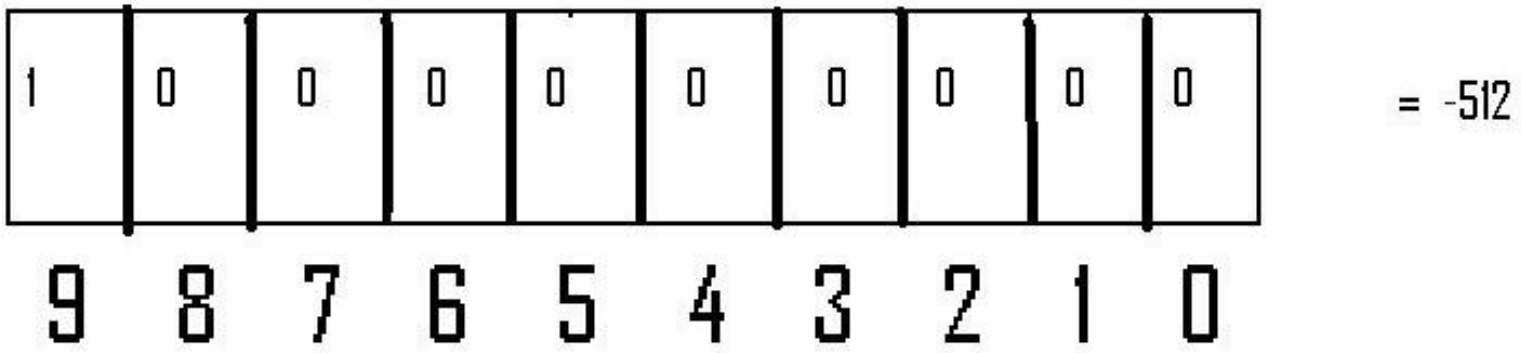


Figura: Máximo entero



Los enteros

Los enteros, almacenados en 10 bits eran interpretados como binarios en complemento a dos: el más significativo en 1 significa -512.
Por lo tanto su interpretación es

$$-512*b_9+256*b_8+128*b_7+64*b_6+32*b_5+16*b_4+8*b_3+4*b_2+2*b_1+b_0$$

donde b_i son los bits de la palabra.

El texto recopila la presentación

Las instrucciones

- Las instrucciones son unidades de información de 20 bits

Las instrucciones

- Las instrucciones son unidades de información de 20 bits
- los primeros 7 son para identificar la función u operación, 3 para indicar el número de registro B-modificador y 10 para indicar la dirección del operando o en algunos casos el operando mismo.

Las instrucciones

- Las instrucciones son unidades de información de 20 bits
- los primeros 7 son para identificar la función u operación, 3 para indicar el número de registro B-modificador y 10 para indicar la dirección del operando o en algunos casos el operando mismo.
- pero para no escribir en binario, las instrucciones tenian 2 dígitos decimales para el código de operación y una función de traducción

Las instrucciones

Las instrucciones ocupan 20 bits. Los diez primeros para la operación, divididos en código de operación, los 7 primeros y los restantes para indicar el registro donde se guardaba la cantidad sumada al contenido de los siguiente 10 bits. Estos últimos 10 bits se interpretaban como la dirección de un operando o como el valor del operando.
Los registros se denominaban B0...B7, siendo B0 un registro cuyo valor era siempre cero (un uso para este registro era la interpretación de ser el registro que almacenaba la constante cero). Las b-modificaciones permitían trabajar con vectores o con direcciones relativas.

El texto recopila la presentación

Las instrucciones

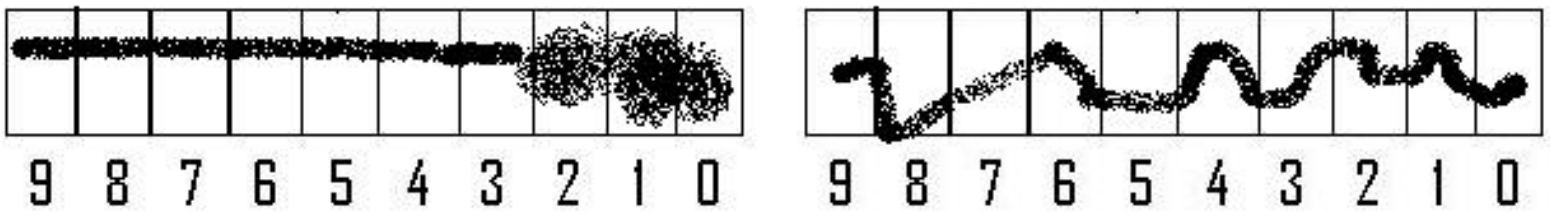


Figura: los bits de la instrucción

Las instrucciones

- 0 . . . 990 parar

Las instrucciones

- 0 . . . 990 parar
- 160 . . . 680 traer la página seleccionada del tambor a la página de memoria

Las instrucciones

- 0 . . . 990 parar
- 160 . . . 680 traer la página seleccionada del tambor a la página de memoria
- En las diversas posiciones de una página podía corresponder a instrucciones, a enteros o a valores reales

Las instrucciones

- Se aprendió como se hacía un compilador leyendo el compilador del AUTOCODE

Las instrucciones

- Se aprendió como se hacía un compilador leyendo el compilador del AUTOCODE
- Lo que puede llamarse hoy "Lectura de los clásicos"

Las instrucciones

- Se aprendió como se hacía un compilador leyendo el compilador del AUTOCODE
- Lo que puede llamarse hoy “Lectura de los clásicos”
- Pero se debía leer el compilador a partir del vuelco de memoria

Los vuelcos de memoria

- Programa de impresión de páginas

Los vuelcos de memoria

- Programa de impresión de páginas
  - Interpretada como (instrucciones, operando)

Los vuelcos de memoria

- Programa de impresión de páginas
  - Interpretada como (instrucciones, operando)
  - Interpretada como (entero, entero)

Los vuelcos de memoria

- Programa de impresión de páginas
  - Interpretada como (instrucciones, operando)
  - Interpretada como (entero, entero)
  - Interpretada como constante flotante

Los vuelcos de memoria

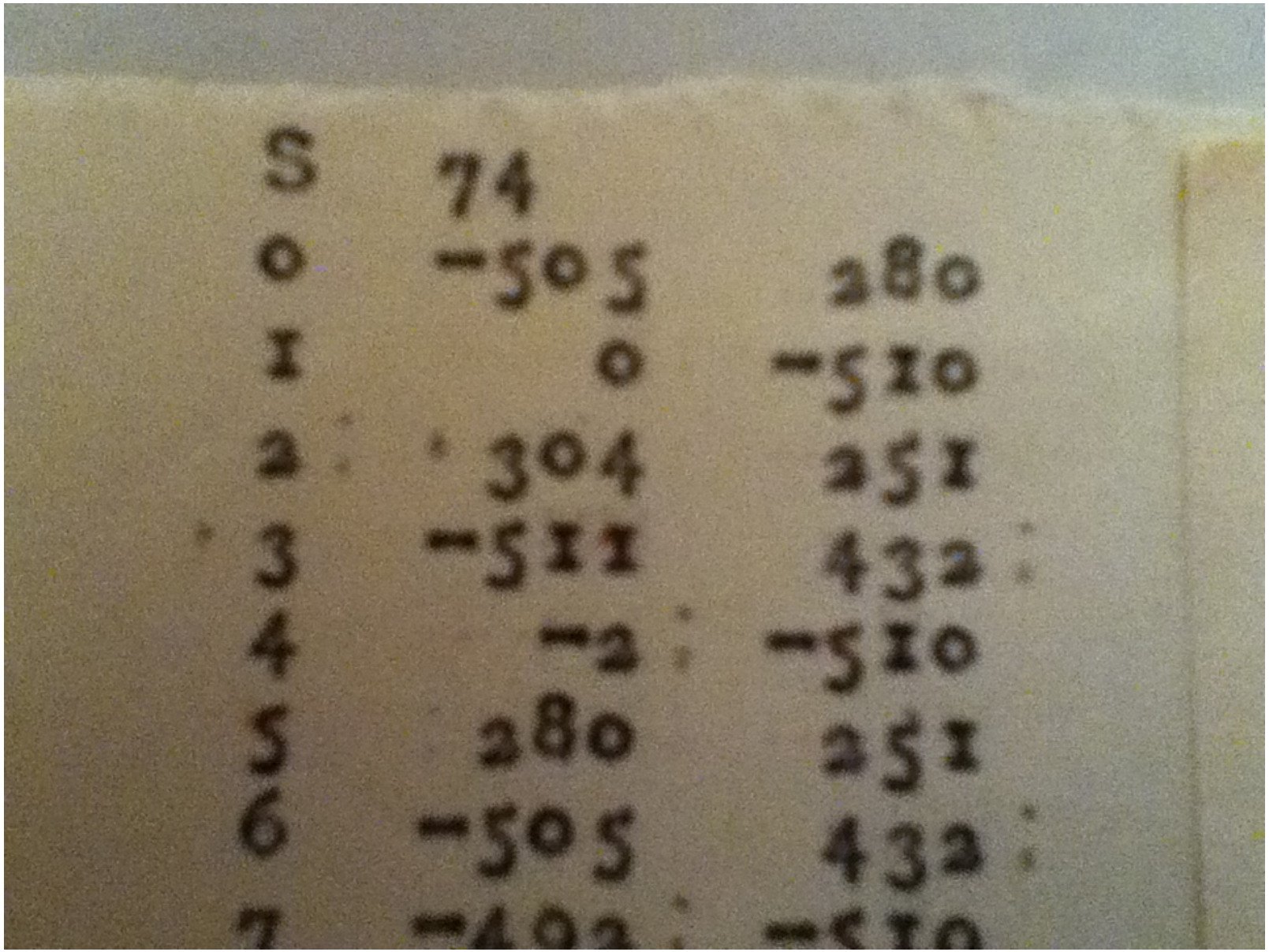
Una característica del código es que residía en los tambores y éstos estaban divididos en sectores.
Otra característica del código era que las constantes, las zonas de variables y las instrucciones estaban mezcladas. Felizmente existía un programa que imprimía sectores del tambor, haciendo una interpretación homogénea de su contenido:

- Entero
- Número en punto flotante
- Instrucción

Es por eso que un cierto sector era interpretado de una única forma como puede verse en la transparencia siguiente donde el sector 74 interpretado como enteros.

El texto recopila la presentación

Los post-mortem



Los post-mortem

Este mismo sector 74 se imprimía como instrucciones, con la salvedad que el assembler tenía códigos nemónicos (si bien numéricos) para codificar las instrucciones. Los dos primeros dígitos para el código de operación y el tercero para la b-modificación. En este modo de impresión los últimos 10 bits se interpretan de acuerdo con el código de operación: número de página si era una operación de traer una página desde el tambor, entero si era una operación de operando inmediato de enteros, página punto línea en caso de direcciones.
Los primeros 10 bits que no podían ser interpretados como instrucciones se indicaban con un = seguido de la interpretación como entero de 10 bits. Por ejemplo, la línea 1 que es interpretada en instrucciones como 990 -510 corresponde a una secuencia de 10 ceros seguido de un 1, seguida de siete ceros, seguida de 10. En la impresión de enteros se imprime el par 0, -510.

El texto recopila la presentación

Los post-mortem

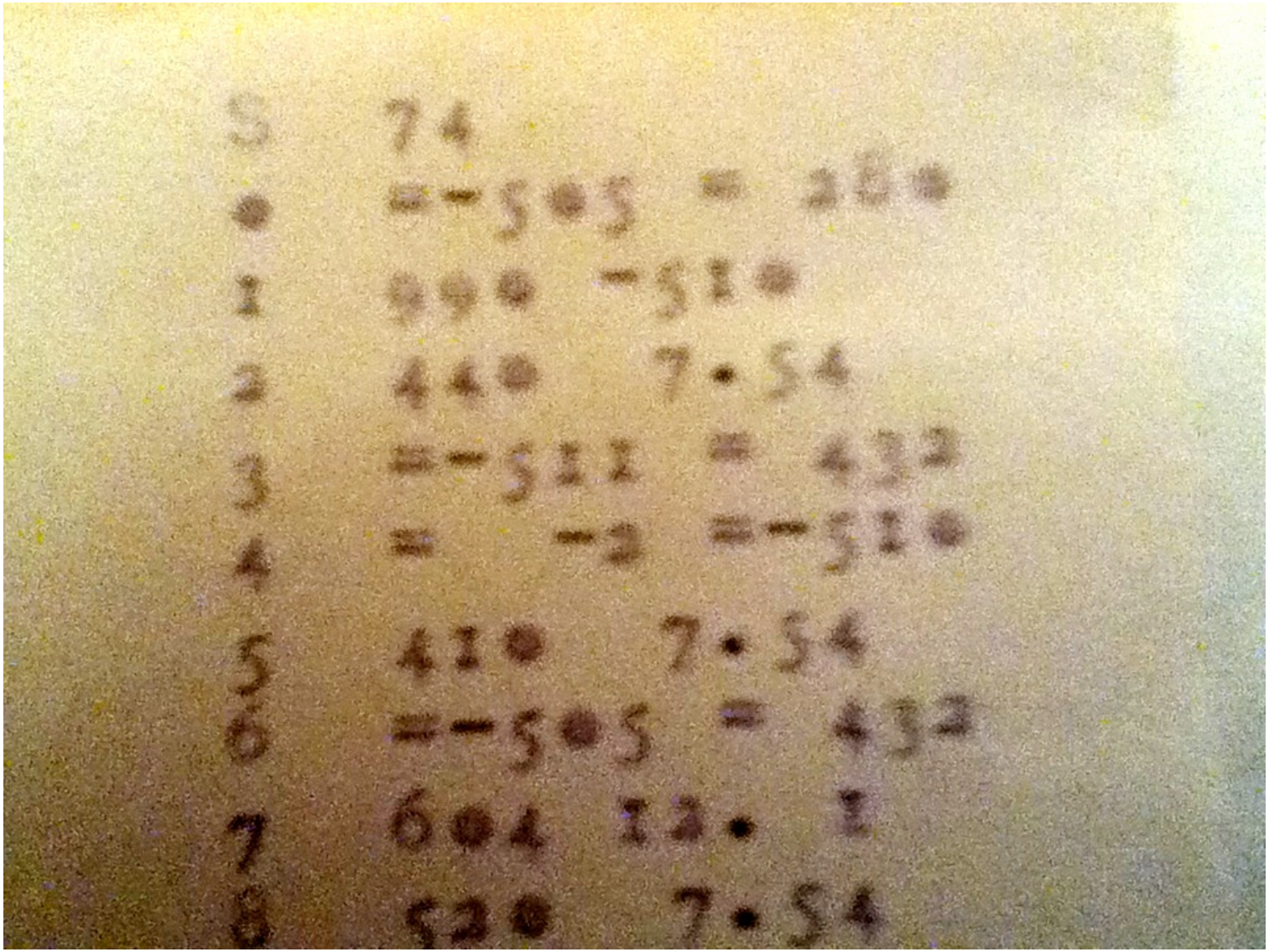
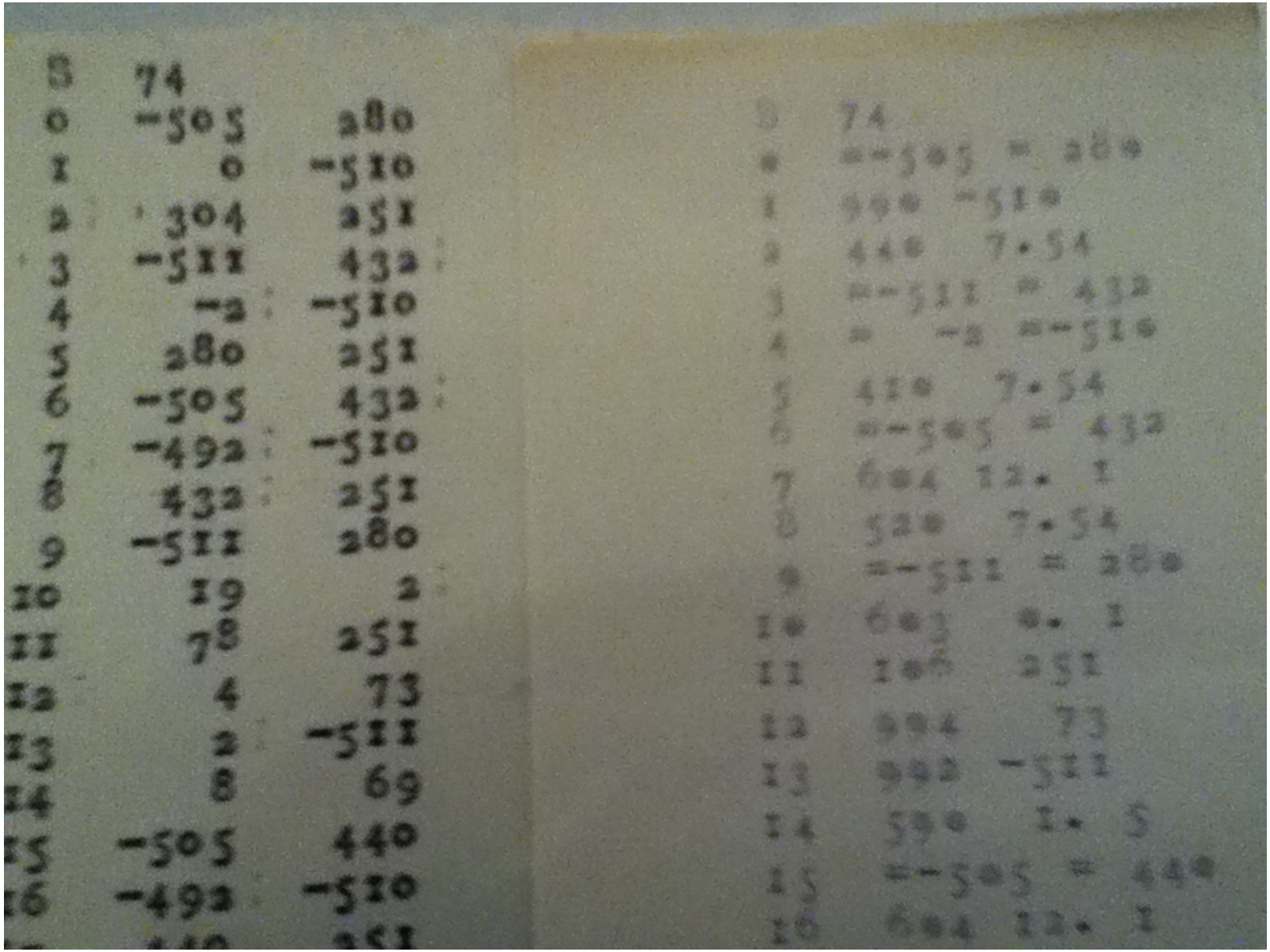


Figura: Sector 74 como instrucciones

Los post-mortem



Los post-mortem

Una vez obtenidos los post-mortem de cada uno los sectores del compilador en sus tres versiones, venía el momento del

CUT & PASTE

Leyendo el código debía asignársele el tipo a cada palabra. Si esa palabra no era del tipo de la página resultado se debía elegir la página de impresión correspondiente, recortar y pegar en la página resultado.
Una vez terminado el pegado, se marcaban con colores las rupturas de control. Estas se producían por una instrucción de bifurcación, o porque la instrucción de traer un sector era para traerla a la misma página que se estaba ejecutando, lo que producía que la próxima instrucción era la del sector recién ingresado en memoria.

El texto recopila la presentación

Los post-mortem

Luego de la completación de las manualidades, disponiamos del código fuente.
Cada miembro del equipo poseía su copia personal (hecha a mano, las fotocopias no existían y mucho mas difícil era reproducir el color).
Durante el trabajo de inferir tipos y ponerle colores a las instrucciones, aprendimos muchos de los mecanismos usados en la compilación. Por ejemplo la pila.
Pese a buscar esos ejemplares no hemos podido encontrar ninguno.

El texto recopila la presentación

Esquema de la presentación

Modelo ganadero

Por alguna circunstancia que no recuerdo bien, se me encargó la programación de un modelo ganadero, diseñado por Oscar Varsavsky.
Era el primer programa de cierta talla que yo programaba en Comic. Como siempre en la programación uno hace análisis de resultados de forma que no estén en franca diferencia con lo esperado.
En ese momento había dos fuentes importantes de error: El código del programa y el lenguaje de programación, ya que era nuevo.
Cuando Oscar Varsavky estimó que el modelo debía estar listo se asomó por la oficina del equipo formado por Wilfred Durán (el jefe) , Clarisa Cortés y yo. Le informé que la programación no estaba lista y él se alejó un poco decepcionado.
La escena se repitió varias veces, mientras que yo, entre visita y visita, revisaba desesperadamente el programa en busca de la fuente de error, ya que me parecía que éste no producía resultados admisibles.

El texto recopila la presentación

Modelo ganadero

El modelo pretendía hacer una proyección en el tiempo del stock ganadero nacional.
En una de esas visitas, Varsavky no admitió un "no está listo" por respuesta y empezó a decir que no podía comprender como me tardaba tanto. Le expliqué que infortunadamente el programa estaba dando resultados erróneos: **el stock era negativo**.
Ante mi sorpresa, puso una gran sonrisa y dijo algo parecido a esto: Era exactamente lo que pretendía mostrar con el modelo. Si se mantenía política ganadera de ese gobierno (1966) el stock ganadero desaparecería!!!

El texto recopila la presentación

Fin

Gracias

Agradezco la colaboración de C. Cortés, W. Durán, N. García, M. Larramendy y L. Lew en la revisión del texto.