Optimal Offset Assignment

Roberto Castañeda Lozano

rcas@sics,kth.se
Offset Assignment

- Classic compiler problem (> 20 years)
- Reorder variables stored in memory
  - minimize their address computation overhead
- Applies to processors with autoincrement mode:
  
  \[
  \begin{align*}
  &= \text{MEM}[r] \\
  &= \text{MEM}[r++] \\
  &= \text{MEM}[r--]
  \end{align*}
  
- reduces program size, popular in signal processors
Offset Assignment: Example

\[ r = &a = \text{MEM}[r] \]
\[ r = &b = \text{MEM}[r] \]
\[ r = &d = \text{MEM}[r] \]
\[ r = &b = \text{MEM}[r] \]
\[ r = &c = \text{MEM}[r] \]
\[ r = &a = \text{MEM}[r] \]
Offset Assignment: Example

\[ r = \&a \]

\[ = \text{MEM}[r] \]
Offset Assignment: Example

\[ r = \&a \]
\[ = \text{MEM}[r] \]

\[ r = \&b \]
\[ = \text{MEM}[r] \]
Offset Assignment: Example

\[ r = \&a \]
\[ \quad = \text{MEM}[r] \]
\[ r = \&b \]
\[ \quad = \text{MEM}[r] \]
\[ r = \&d \]
\[ \quad = \text{MEM}[r] \]
Offset Assignment: Example

\[
\begin{align*}
r & = \&a \\
& = \text{MEM}[r] \\
r & = \&b \\
& = \text{MEM}[r] \\
r & = \&d \\
& = \text{MEM}[r] \\
r & = \&b \\
& = \text{MEM}[r]
\end{align*}
\]
Offset Assignment: Example

\[ r = \&a \]
\[ = \text{MEM}[r] \]
\[ r = \&b \]
\[ = \text{MEM}[r] \]
\[ r = \&d \]
\[ = \text{MEM}[r] \]
\[ r = \&b \]
\[ = \text{MEM}[r] \]
\[ r = \&c \]
\[ = \text{MEM}[r] \]
Offset Assignment: Example

\[
r = &a \\
= \text{MEM}[r] \\
r = &b \\
= \text{MEM}[r] \\
r = &d \\
= \text{MEM}[r] \\
r = &b \\
= \text{MEM}[r] \\
r = &c \\
= \text{MEM}[r] \\
r = &a \\
= \text{MEM}[r] \\
\]
Offset Assignment: Example

\[ r = &a \]
\[ = \text{MEM}[r] \]

\[ r = &b \]
\[ = \text{MEM}[r] \]

\[ r = &d \]
\[ = \text{MEM}[r] \]

\[ r = &b \]
\[ = \text{MEM}[r] \]

\[ r = &c \]
\[ = \text{MEM}[r] \]

\[ r = &a \]
\[ = \text{MEM}[r] \]

- 6 address loads
Offset Assignment: Autoincrement

\[
\begin{align*}
    r &= \&a \\
    &= \text{MEM}[r] \\
    r &= \&b \\
    &= \text{MEM}[r] \\
    r &= \&d \\
    &= \text{MEM}[r] \\
    r &= \&b \\
    &= \text{MEM}[r] \\
    r &= \&c \\
    &= \text{MEM}[r] \\
    r &= \&a \\
    &= \text{MEM}[r]
\end{align*}
\]
Offset Assignment: Autoincrement

\[
\begin{align*}
    r &= \&a \\
    &= \text{MEM}[r++] \\
    r &= \&b \\
    &= \text{MEM}[r] \\
    r &= \&d \\
    &= \text{MEM}[r] \\
    r &= \&b \\
    &= \text{MEM}[r] \\
    r &= \&c \\
    &= \text{MEM}[r] \\
    r &= \&a \\
    &= \text{MEM}[r]
\end{align*}
\]
Offset Assignment: Autoincrement

\[ r = \&a \]

\[ = \text{MEM}[r++] \]

\[ r = \&b \]

\[ = \text{MEM}[r] \]

\[ r = \&d \]

\[ = \text{MEM}[r] \]

\[ r = \&b \]

\[ = \text{MEM}[r] \]

\[ r = \&c \]

\[ = \text{MEM}[r] \]

\[ r = \&a \]

\[ = \text{MEM}[r] \]
Offset Assignment: Autoincrement

\[ r = \&a \]
\[ = \text{MEM}[r++] \]
\[ r = \&b \]
\[ = \text{MEM}[r] \]
\[ r = \&d \]
\[ = \text{MEM}[r] \]
\[ r = \&b \]
\[ = \text{MEM}[r++] \]
\[ r = \&c \]
\[ = \text{MEM}[r] \]
\[ r = \&a \]
\[ = \text{MEM}[r] \]
Offset Assignment: Autoincrement

\[ r = \&a \]
\[ = \text{MEM}\[r++\] \]
\[ r = \&b \]
\[ = \text{MEM}\[r]\] \]
\[ r = \&d \]
\[ = \text{MEM}\[r]\] \]
\[ r = \&b \]
\[ = \text{MEM}\[r++\] \]
\[ r = \&c \]
\[ = \text{MEM}\[r]\] \]
\[ r = \&a \]
\[ = \text{MEM}\[r]\] \]
Offset Assignment: Autoincrement

\[
\begin{align*}
  r &= \&a \\
      &= \text{MEM}[r++] \\
  r &= \&b \\
      &= \text{MEM}[r] \\
  r &= \&d \\
      &= \text{MEM}[r] \\
  r &= \&b \\
      &= \text{MEM}[r++] \\
  r &= \&c \\
      &= \text{MEM}[r] \\
  r &= \&a \\
      &= \text{MEM}[r]
\end{align*}
\]

- 4 address loads
Offset Assignment: Better Memory Layout

- `r = &a` = `MEM[r++]`
- `r = &b` = `MEM[r++]`
- `r = &d` = `MEM[r--]`
- `r = &b` = `MEM[r]`
- `r = &c` = `MEM[r]`
- `r = &a` = `MEM[r]`
Offset Assignment: Better Memory Layout

\[ r = \&a \]
\[ = \text{MEM}[r++] \]

\[
\begin{array}{c}
a \\
b \\
d \\
c
\end{array}
\]
Offset Assignment: Better Memory Layout

\[ r = \&a \]
\[ = \text{MEM}[r++] \]

\[ r = \&b \]
\[ = \text{MEM}[r++] \]

\[ r = \&d \]
\[ = \text{MEM}[r--] \]

\[ r = \&b \]
\[ = \text{MEM}[r] \]

\[ r = \&c \]
\[ = \text{MEM}[r] \]

\[ r = \&a \]
\[ = \text{MEM}[r] \]
Offset Assignment: Better Memory Layout

\[
\begin{align*}
    r &= \&a \\
    &= \text{MEM}[r++] \\
    r &= \&b \\
    &= \text{MEM}[r++] \\
    r &= \&d \\
    &= \text{MEM}[r--]
\end{align*}
\]
Offset Assignment: Better Memory Layout

r = &a
= MEM[r++]

r = &b
= MEM[r++]

r = &d
= MEM[r--]

r = &b
= MEM[r]
Offset Assignment: Better Memory Layout

\[
\begin{align*}
  r &= \&a \\
  &= \text{MEM}[r++] \\
  r &= \&b \\
  &= \text{MEM}[r++] \\
  r &= \&d \\
  &= \text{MEM}[r--] \\
  r &= \&b \\
  &= \text{MEM}[r] \\
  r &= \&c \\
  &= \text{MEM}[r]
\end{align*}
\]
Offset Assignment: Better Memory Layout

\[
\begin{align*}
    \text{r} &= & \&\text{a} \\
    &= & \text{MEM}[\text{r}++] \\
    \text{r} &= & \&\text{b} \\
    &= & \text{MEM}[\text{r}++] \\
    \text{r} &= & \&\text{d} \\
    &= & \text{MEM}[\text{r}--] \\
    \text{r} &= & \&\text{b} \\
    &= & \text{MEM}[\text{r}] \\
    \text{r} &= & \&\text{c} \\
    &= & \text{MEM}[\text{r}] \\
    \text{r} &= & \&\text{a} \\
    &= & \text{MEM}[\text{r}] 
\end{align*}
\]
Offset Assignment: Better Memory Layout

\[
\begin{align*}
r &= \&a \\
    &= \text{MEM}[r++] \\
r &= \&b \\
    &= \text{MEM}[r++] \\
r &= \&d \\
    &= \text{MEM}[r--] \\
r &= \&b \\
    &= \text{MEM}[r] \\
r &= \&c \\
    &= \text{MEM}[r] \\
r &= \&a \\
    &= \text{MEM}[r]
\end{align*}
\]

- 3 address loads
Offset Assignment: Optimal Approach

- Access Graph: transitions between variables
- Example for sequence (a, b, d, b, c, a):

Reduce to Traveling Salesman Problem (TSP) [Jünger and Mallach, 2013]
- Integer Programming (IP) approach
Offset Assignment: Optimal Approach

- Access Graph: transitions between variables
- Example for sequence (a, b, d, b, c, a):

Reduce to Traveling Salesman Problem (TSP) [Jünger and Mallach, 2013]

Integer Programming (IP) approach
General Offset Assignment

- Generalization to multiple address registers $r_1, r_2, \ldots$
- Two interdependent subproblems:
  - Offset Assignment: which ordering of vars in memory?
  - Register Assignment: which reg for each access?
General Offset Assignment: Example

\[ r_1 = &a \]
\[ r_1 = &b \]
\[ r_2 = &d \]
\[ r_1 = &b \]
\[ r_2 = &c \]
\[ r_1 = &a \]

\[ \text{2 address loads} \]

\[ a \]
\[ b \]
\[ c \]
\[ d \]
General Offset Assignment: Example

\[ r1 = \&a \]

\[ = \text{MEM}[r1++] \]

\[ r1 = \&b \]

\[ r2 = \&d \]

\[ r1 = \&b \]

\[ r2 = \&c \]

\[ r1 = \&a \]
General Offset Assignment: Example

\[
\begin{align*}
  r1 &= \&a \\
  &= \text{MEM}[r1++] \\
  r1 &= \&b \\
  &= \text{MEM}[r1] \\
  r2 &= \&d \\
  &= \text{MEM}[r2--] \\
  r1 &= \&b \\
  &= \text{MEM}[r1--] \\
  r2 &= \&c \\
  &= \text{MEM}[r2] \\
  r1 &= \&a \\
  &= \text{MEM}[r1] \\
\end{align*}
\]
General Offset Assignment: Example

\[
\begin{align*}
    r1 &= \&a \\
        &= \text{MEM}[r1++] \\
    r1 &= \&b \\
        &= \text{MEM}[r1] \\
    r2 &= \&d \\
        &= \text{MEM}[r2--] \\
\end{align*}
\]
General Offset Assignment: Example

\[
r1 = \&a = \text{MEM}[r1++]
\]

\[
r1 = \&b = \text{MEM}[r1]
\]

\[
r2 = \&d = \text{MEM}[r2--]
\]

\[
r1 = \&b = \text{MEM}[r1--]
\]
General Offset Assignment: Example

\begin{align*}
\text{r1} &= \&a \\
&= \text{MEM}[\text{r1}++] \\
\text{r1} &= \&b \\
&= \text{MEM}[\text{r1}] \\
\text{r2} &= \&d \\
&= \text{MEM}[\text{r2}--] \\
\text{r1} &= \&b \\
&= \text{MEM}[\text{r1}--] \\
\text{r2} &= \&c \\
&= \text{MEM}[\text{r2}] 
\end{align*}
General Offset Assignment: Example

\[
\begin{align*}
r1 &= \&a \\
&= \text{MEM}[r1++] \\
r1 &= \&b \\
&= \text{MEM}[r1] \\
r2 &= \&d \\
&= \text{MEM}[r2--] \\
r1 &= \&b \\
&= \text{MEM}[r1--] \\
r2 &= \&c \\
&= \text{MEM}[r2] \\
r1 &= \&a \\
&= \text{MEM}[r1]
\end{align*}
\]
General Offset Assignment: Example

\[ r1 = \&a \]
\[ = \text{MEM}[r1++] \]
\[ r1 = \&b \]
\[ = \text{MEM}[r1] \]
\[ r2 = \&d \]
\[ = \text{MEM}[r2--] \]
\[ r1 = \&b \]
\[ = \text{MEM}[r1--] \]
\[ r2 = \&c \]
\[ = \text{MEM}[r2] \]
\[ r1 = \&a \]
\[ = \text{MEM}[r1] \]

\( 2 \text{ address loads} \)
General Offset Assignment: Optimal Approach

- Optimal Register Assignment as a network flow problem [Gebotys 1997]
- For \((a_1, b_1, d, b_2, c, a_2)\), assuming offset \((a, b, c, d)\):

![Diagram of the network flow problem](image_url)
General Offset Assignment: Optimal Approach

- Optimal Register Assignment as a network flow problem [Gebotys 1997]
- For \((a_1, b_1, d, b_2, c, a_2)\), assuming offset \((a, b, c, d)\):
General Offset Assignment: Optimal Approach

- Optimal Register Assignment as a network flow problem [Gebotys 1997]
- For \((a_1, b_1, d, b_2, c, a_2)\), assuming offset \((a, b, c, d)\):

- IP approach that integrates:
  - TSP model (offset assignment)
  - flow network model (register assignment)

[Mallach and Castañeda, 2014]