

Implementation

Examples



Pirouz Bazargan Sabet

December 2008

Example

Compiler's rule

function's arguments are put on a stack

R29 : stack pointer

R2 : returned value

R5 to R26 : local variables



Pirouz Bazargan Sabet

December 2008

Example

```
int strlen (char *s)
{
    int i = 0;
    while (s[i] != '\0')
        i++;
    return (i)
}
```



Pirouz Bazargan Sabet

December 2008

Example

Classic compiler

```
int strlen (char *s)
{
    int i = 0;
    while (s[i] != '\0')
        i++;
    return (i)
}
```

Lw	R5	, 0 (R29)
Add	R2	, R0 , R0
_While:	Lb	R6 , 0 (R5)
	Beq	R6 , R0 , _End
	Addiu	R5 , R5 , +1
	Addi	R2 , R2 , +1
	J	_While
_End:	Jr	R31



Pirouz Bazargan Sabet

December 2008

Example

Classic compiler for pipeline processor

```

int strlen (char *s)
{
  int i = 0;
  while (s[i] != '\0')
    i++;
  return (i)
}

        Lw   R5 , 0 (R29)
        Add  R2 , R0 , R0
        _While: Lb   R6 , 0 (R5 )
                Beq  R6 , R0 , _End
                NOP
                Addiu R5 , R5 , +1
                Addi R2 , R2 , +1
                J    _While
                NOP
        _End:  Jr   R31
                NOP
    
```



Pirouz Bazargan Sabet

December 2008

Example

Basic compiler for pipeline processor

```

_while: Lb   R6 , 0 (R5 )
        Beq  R6 , R0 , _End
        NOP
        Addiu R5 , R5 , +1
        Addi R2 , R2 , +1
        J    _while
        NOP
    
```



Pirouz Bazargan Sabet

December 2008

9 cycles per element
5 usefull instructions - 4 cycles lost

Example

Classic compiler for pipeline processor

```

int strlen (char *s)
{
  int i = 0;
  while (s[i] != '\0')
    i++;
  return (i)
}

        Lw   R5 , 0 (R29)
        Addi R2 , R0 , -1
        _While: Lb   R6 , 0 (R5 )
                Addiu R5 , R5 , +1
                Addi R2 , R2 , +1
                Bne  R6 , R0 , _While
                NOP
        _End:  Jr   R31
                NOP
    
```



Pirouz Bazargan Sabet

December 2008

Example

Basic compiler for pipeline processor

```

_while: Lb   R6 , 0 (R5 )
        Addiu R5 , R5 , +1
        Addi R2 , R2 , +1
        Bne  R6 , R0 , _while
        NOP
    
```



Pirouz Bazargan Sabet

December 2008

5 cycles per element
4 usefull instructions - 1 cycle lost

Example

Classic compiler for pipeline processor

```

int strlen (char *s)
{
  int i = 0;
  while (s[i] != '\0')
    i++;
  return (i)
}

Lw    R5 , 0 (R29)
Addi  R2 , R5 , +1
_While: Lb    R6 , 0 (R5 )
      Addiu  R5 , R5 , +1
      Bne   R6 , R0 , _While
      NOP
      Sub   R2 , R5 , R2
_End:  Jr    R31
      NOP
  
```

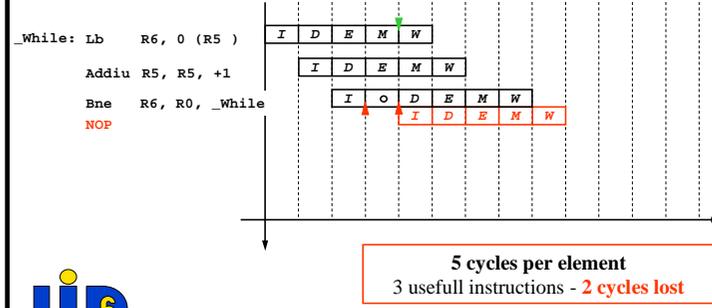


Pirouz Bazargan Sabet

December 2008

Example

Basic compiler for pipeline processor



Pirouz Bazargan Sabet

December 2008

Example

Classic compiler for pipeline processor

```

int strlen (char *s)
{
  int i = 0;
  while (s[i] != '\0')
    i++;
  return (i)
}

_While: Lb    R6 , 0 (R5 )
      Bne   R6 , R0 , _While
      Addiu  R5 , R5 , +1
      Sub   R2 , R5 , R2
_End:  Jr    R31
      NOP
  
```

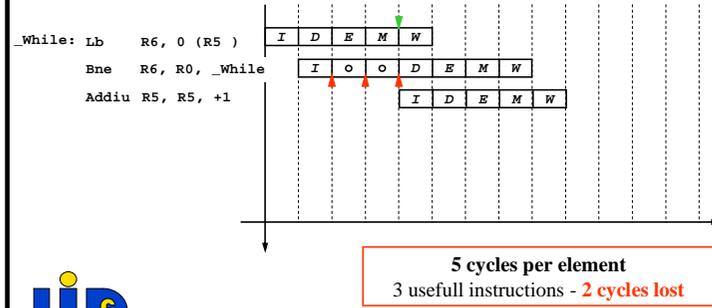


Pirouz Bazargan Sabet

December 2008

Example

Basic compiler for pipeline processor



Pirouz Bazargan Sabet

December 2008

Example

```
void func (int *v, int size)
{
  int i;
  for (i=0 ; i<size ; i++)
    v [i] = 2 * v [i];
}
```



Pirouz Bazargan Sabet

December 2008

Example

Classic compiler

```
void func (int *v, int size)
{
  int i;
  for (i=0 ; i<size ; i++)
    v [i] = 2 * v [i];
}
```

```
Lw   R5 , 0 (R29)
Lw   R6 , 4 (R29)
Add  R7 , R0 , R0
_For: Beq  R7 , R6 , _EndFor

Lw   R8 , 0 (R5 )
Sll  R8 , R8 , 1
Sw   R8 , 0 (R5 )
Addiu R5 , R5 , +4
Addi R7 , R7 , +1

J    _For

_EndFor: Jr   R31
```



Pirouz Bazargan Sabet

December 2008

Example

Basic compiler for pipeline processor

```
void func (int *v, int size)
{
  int i;
  for (i=0 ; i<size ; i++)
    v [i] = 2 * v [i];
}

_For: Beq  R7 , R6 , _EndFor
      NOP
      Lw   R8 , 0 (R5 )
      Sll  R8 , R8 , 1
      Sw   R8 , 0 (R5 )
      Addiu R5 , R5 , +4
      Addi R7 , R7 , +1

      J    _For
      NOP
_EndFor: Jr   R31
        NOP
```

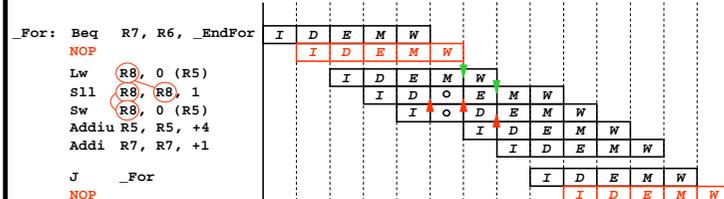


Pirouz Bazargan Sabet

December 2008

Example

Basic compiler for pipeline processor



10 cycles per element
7 usefull instructions - 3 cycles lost



Pirouz Bazargan Sabet

December 2008

Example

Basic compiler for pipeline processor

```

void func (int *v, int size)
{
  int i;
  for (i=0 ; i<size ; i++)
    v [i] = 2 * v [i];
}

```

```

                Lw   R5 , 0 (R29)
                Lw   R6 , 4 (R29)
                Add  R7 , R0 , R0
                Beq  R7 , R6 , _EndFor
                NOP
    _For:        Lw   R8 , 0 (R5 )
                Sll  R8 , R8 , 1
                Sw   R8 , 0 (R5 )
                Addiu R5 , R5 , +4
                Addi R7 , R7 , +1

                Bne  R7 , R6 , _For
                NOP
    _EndFor:    Jr   R31
                NOP

```

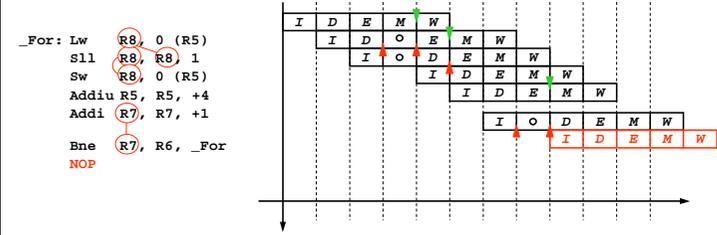


Pirouz Bazargan Sabet

December 2008

Example

Basic compiler for pipeline processor



9 cycles per element
6 usefull instructions - 3 cycles lost



Pirouz Bazargan Sabet

December 2008

Example

Compiler for pipeline processor

```

void func (int *v, int size)
{
  int i;
  for (i=0 ; i<size ; i++)
    v [i] = 2 * v [i];
}

```

```

                Lw   R5 , 0 (R29)
                Lw   R6 , 4 (R29)
                Sll  R9 , R6 , 2
                Add  R9 , R9 , R5
                Beq  R5 , R9 , _EndFor
                NOP
    _For:        Lw   R8 , 0 (R5 )
                Sll  R8 , R8 , 1
                Sw   R8 , 0 (R5 )
                Addiu R5 , R5 , +4

                Bne  R5 , R9 , _For
                NOP
    _EndFor:    Jr   R31
                NOP

```

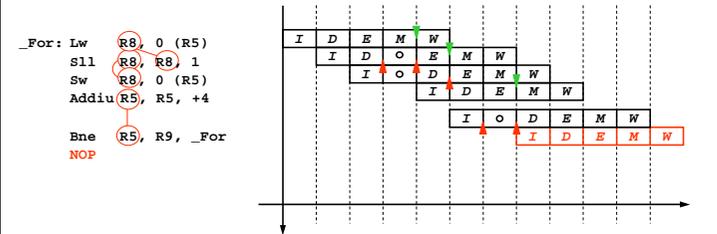


Pirouz Bazargan Sabet

December 2008

Example

Basic compiler for pipeline processor



8 cycles per element
5 usefull instructions - 3 cycles lost



Pirouz Bazargan Sabet

December 2008

Example

Smart compiler for pipeline processor

```

void func (int *v, int size)
{
  int i;
  for (i=0 ; i<size ; i++)
    v [i] = 2 * v [i];
}

```

```

Lw   R5 , 0 (R29)
Lw   R6 , 4 (R29)
Sll  R9 , R6 , 2
Add  R9 , R9 , R5
Beq  R5 , R9 , _EndFor
NOP

_For: Lw   R8 , 0 (R5 )
      Addiu R5 , R5 , +4
      Sll  R8 , R8 , 1
      Bne  R5 , R9 , _For
      Sw   R8 , -4 (R5 )
_EndFor: Jr  R31
NOP

```

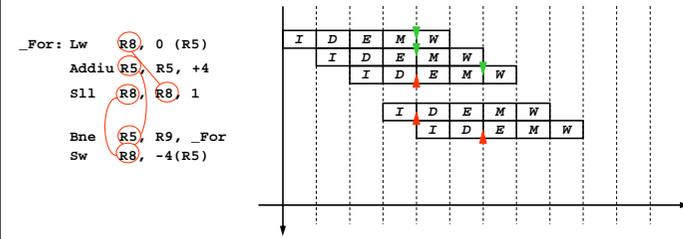


Pirouz Bazargan Sabet

December 2008

Example

Smart compiler for pipeline processor



Pirouz Bazargan Sabet

December 2008

Example

```

void func (int *v, int size)
{
  int i;
  for (i=0 ; i<size ; i++)
    v [i] = 2 * v [i];
}

```

```

Lw   R5 , 0 (R29)
Lw   R6 , 4 (R29)
Sll  R9 , R6 , 2
Add  R9 , R9 , R5
Beq  R5 , R9 , _EndFor
NOP

_For: Lw   R8 , 0 (R5 )
      Addiu R5 , R5 , +4
      Sll  R8 , R8 , 1
      Bne  R5 , R9 , _For
      Sw   R8 , -4 (R5 )
_EndFor: Jr  R31
NOP

```



Pirouz Bazargan Sabet

December 2008

Example

3 instructions to process one element
2 instructions to manage the loop

2 cycles of overhead per element

share the overhead between several elements



Pirouz Bazargan Sabet

December 2008

Example

```
void func (int *v, int size)
{
    int i;
    for (i=0 ; i<size ; i++)
        v [i] = 2 * v [i];
}
```

```

Lw   R5 , 0 (R29)
Lw   R6 , 4 (R29)
Sll  R9 , R6 , 2
Add  R9 , R9 , R5
Beq  R5 , R9 , _EndFor
NOP

_For: Lw   R8 , 0 (R5 )
      Lw   R18, 4 (R5 )
      Addiu R5 , R5 , +8
      Sll  R8 , R8 , 1
      Sll  R18, R18, 1
      Sw   R8 , -8 (R5 )
      Bne  R5 , R9 , _For
      Sw   R18, -4 (R5 )
_EndFor: Jr  R31
NOP
```



Pirouz Bazargan Sabet

December 2008

Example

6 instructions to process two elements

2 instructions to manage the loop

4 cycles per element

Loop Unrolling :

Reduce the overhead

Increase reordering opportunities



Pirouz Bazargan Sabet

December 2008