COMP 412, Lab 3
Early Release Example
Block report04.i

Scheduled Code without Early Release

```
[ loadl 1028 => r16      ; loadl 1024 => r17 ]
[ load  r16 => r10       ; loadl 1032 => r0 ]
[ load  r17 => r15       ; loadl 1036 => r12 ]
[ loadl 1040 => r11      ; loadl 1044 => r6 ]
[ nop                    ; nop ]
[ nop                    ; nop ]
[ nop                    ; nop ]
[ store r15 => r0        ; add  r15, r10 => r14 ]
[ add  r14, r10 => r13   ; nop ]
[ nop                    ; nop ]
[ nop                    ; nop ]
[ store r14 => r12       ; nop ]
[ nop                    ; nop ]
[ nop                    ; nop ]
[ nop                    ; nop ]
[ store r13 => r11       ; nop ]
[ nop                    ; nop ]
[ nop                    ; nop ]
[ nop                    ; nop ]
[ store r10 => r6        ; nop ]
[ nop                    ; nop ]
[ nop                    ; nop ]
[ nop                    ; nop ]
[ load  r0 => r9         ; nop ]
[ load  r12 => r8        ; nop ]
[ load  r11 => r5        ; nop ]
[ load  r6 => r3         ; nop ]
[ nop                    ; nop ]
[ lshift r9, r10 => r7   ; nop ]
[ nop                    ; mult r7, r8 => r4 ]
[ nop                    ; nop ]
[ nop                    ; mult r4, r5 => r2 ]
[ nop                    ; nop ]
[ nop                    ; nop ]
[ nop                    ; mult r2, r3 => r1 ]
[ nop                    ; nop ]
[ nop                    ; nop ]
[ store r1 => r0         ; nop ]
[ nop                    ; nop ]
[ nop                    ; nop ]
[ nop                    ; nop ]
[ output 1032            ; nop ]
```

lab3_ref -e -s -n

Scheduled Code with Early Release

```
[ loadl 1028 => r16      ; loadl 1024 => r17 ]
[ load  r16 => r10       ; loadl 1032 => r0 ]
[ load  r17 => r15       ; loadl 1036 => r12 ]
[ loadl 1040 => r11      ; loadl 1044 => r6 ]
[ nop                    ; nop ]
[ nop                    ; nop ]
[ nop                    ; nop ]
[ store r15 => r0        ; add  r15, r10 => r14 ]
[ store r14 => r12       ; add  r14, r10 => r13 ]
[ store r13 => r11       ; nop ]
[ store r10 => r6        ; nop ]
[ nop                    ; nop ]
[ nop                    ; nop ]
[ nop                    ; nop ]
[ load  r0 => r9         ; nop ]
[ load  r12 => r8        ; nop ]
[ load  r11 => r5        ; nop ]
[ load  r6 => r3         ; nop ]
[ nop                    ; nop ]
[ lshift r9, r10 => r7   ; nop ]
[ nop                    ; mult r7, r8 => r4 ]
[ nop                    ; nop ]
[ nop                    ; mult r4, r5 => r2 ]
[ nop                    ; nop ]
[ nop                    ; nop ]
[ nop                    ; mult r2, r3 => r1 ]
[ nop                    ; nop ]
[ nop                    ; nop ]
[ store r1 => r0         ; nop ]
[ nop                    ; nop ]
[ nop                    ; nop ]
[ nop                    ; nop ]
[ nop                    ; nop ]
[ output 1032            ; nop ]
```

lab3_ref -s -n

Sched. Code w/Memory Disambiguation

```
[ loadl 1028 => r16      ; loadl 1024 => r17 ]
[ load  r16 => r10       ; loadl 1032 => r0 ]
[ load  r17 => r15       ; loadl 1036 => r12 ]
[ loadl 1040 => r11      ; loadl 1044 => r6 ]
[ nop                    ; nop ]
[ nop                    ; nop ]
[ store r10 => r6        ; nop ]
[ store r15 => r0        ; add  r15, r10 => r14 ]
[ store r14 => r12       ; add  r14, r10 => r13 ]
[ store r13 => r11       ; nop ]
[ nop                    ; nop ]
[ load  r6 => r3         ; nop ]
[ load  r0 => r9         ; nop ]
[ load  r12 => r8        ; nop ]
[ load  r11 => r5        ; nop ]
[ nop                    ; nop ]
[ nop                    ; nop ]
[ lshift r9, r10 => r7   ; nop ]
[ nop                    ; mult r7, r8 => r4 ]
[ nop                    ; nop ]
[ nop                    ; nop ]
[ nop                    ; mult r4, r5 => r2 ]
[ nop                    ; nop ]
[ nop                    ; nop ]
[ nop                    ; mult r2, r3 => r1 ]
[ nop                    ; nop ]
[ nop                    ; nop ]
[ store r1 => r0         ; nop ]
[ nop                    ; nop ]
[ nop                    ; nop ]
[ nop                    ; nop ]
[ output 1032            ; nop ]
```

lab3_ref (no options)

Notes:

1. Example shows the code after renaming. The names are consistent across all three examples of scheduled code, but they differ from the source code.

2. Early release saves 12 cycles out of 48 (or 25%) on this block.

3. Adding constant-based disambiguation of loads and stores to early release saves an additional 3 cycles out of 36 (or 8.3%) on this block.