



h3

 main[Browse files](#) jalcant4 committed last week

1 parent d10e61e commit 7c32b85

Showing 1 changed file with 114 additions and 8 deletions.

[Split](#) [Unified](#)

▼ 122 ■■■■■ hv3/mips1.asm 			...		
@@ -47,7 +47,9 @@					
47	.align 2		47	.align 2	
48	LINES: .word 0:10 # an array of 10 integers(words), each is initialized to		48	LINES: .word 0:10 # an array of 10 integers(words), each is initialized to	
49	be 0		49	be 0	
50	- DEST: .word 99:10		50	+ DEST: .word 99:10	
51	hex: .byte		51	+ DIFF: .word 32:10	
52	'0','1','2','3','4','5','6','7','8','9','A','B','C','D','E','F'		52		
53	# Feel free to define more data elements		53	hex: .byte	
			54	'0','1','2','3','4','5','6','7','8','9','A','B','C','D','E','F'	
			55	# Feel free to define more data elements	
@@ -150,27 +152,35 @@ main:					
150	la \$s3, DEST		152	la \$s3, DEST	
151	addi \$s2, \$0, 1 #instruction cycle counter		153	addi \$s2, \$0, 1 #instruction cycle counter	
152	addi \$s4, \$0, 1 #loop incremter		154	addi \$s4, \$0, 1 #loop incremter	
153	main_loop:		155	+ la \$4, NEWLINE #print newline	
154	addi \$s6, \$0, 0		156	+ jal print_string	
155	- la \$4, NEWLINE #print newline		157	main_loop:	
156	- jal print_string		158	addi \$s6, \$0, 0	
157	la \$4, INSN_HEAD #print instruction head		159	la \$4, INSN_HEAD #print instruction head	
158	jal print_string		160	jal print_string	
159	print_int(\$s4) #print instruction head such that I1, I2, ... IN		161	print_int(\$s4) #print instruction head such that I1, I2, ... IN	
160	- la \$4, SRCREGS		162	+ la \$4, SRCREGS # defence info	
161	jal print_string		163	jal print_string	
162	-		164	+ addi \$t3, \$s4, -1 # isn - 1	
163	addi \$t3, \$s4, -1 # isn - 1		165	sll \$t3, \$t3, 2	
164	sll \$t3, \$t3, 2		166	add \$a0, \$s1, \$t3 # add the isn to the base address to get a0 =	
165	add \$a0, \$s1, \$t3 # add the isn to the base address to get a0 =		167	LINES[isn - 1], machine code	
166	LINES[isn - 1], machine code		168	lw \$a0, 0(\$a0) # extracts the machine code from the array	
167	lw \$a0, 0(\$a0) # extracts the machine code from the array		169	jal get_regs # gets the source and desination register of the	
168	jal get_regs # gets the source and desination register of the			machine code, a0 first src, a1 second src, a2 dest	
169	machine code, a0 first src, a1 second src, a2 dest		170	+ addi \$v1, \$a1, 0	
170	addi \$v1, \$a1, 0		171	jal compare # compares the source register with the previous	
171	jal compare # compares the source register with the previous		172	desination registers to see if there is a dependency	
172	- sub \$t8, \$t8, \$s2		173	+ addi \$a1, \$v1, 0	
173	- sub \$t9, \$t9, \$s2		174	jal compare2 # compares the second source register with the	
174	array sw \$a2, 0(\$s3) # saves the destination register into the dest		175	previous desination registers to see if there is a dependency	
175	addi \$s3, \$s3, 4 # offset of next array item		176	+ sub \$a0, \$t8, \$0 # a0 = instruction 1	
176			177	+ sub \$a1, \$t9, \$0 # a1 = instruction 2	
			178	+ jal get_start_cycle	
			179	+ add \$a0, \$v0, \$0 # add the result of start cycle	
			180	+ jal print_cycle	
			181	+ sw \$a2, 0(\$s3) # saves the destination register into the dest	
			182	+ addi \$s3, \$s3, 4 # offset of next array item	
			183	+ array	
			184	addi \$s3, \$s3, 4 # offset of next array item	
			185		
			186		
@@ -332,6 +342,91 @@ print_cycle:					
332			342	##### Helper Method Section #####	
333	##### Helper Method Section #####		343	#####	
334	#####		344	#####	
			345	+ # this method prints the start cycle	
			346	+ # arg a0 compare 1 result	
			347	+ # arg a1 compare 2 result	
			348	+ # ret v0 start cycle	
			349	+ #####	
			350	+ .globl get_start_cycle	
			351	+ get_start_cycle:	
			352	+ addi \$sp, \$sp, -4 #save argument and \$ra	
			353	+ sw \$ra, 0(\$sp)	
			354	+ addi \$t1, \$0, 1 # check if I1	
			355	+ bne \$t1, \$s4, i2_onward # start comparing instructions after I1	
			356	+ la \$t0, DIFF # difference holds the instruction start cycle of each	
			357	+ instruction	
			358	+ addi \$t1, \$s4, -1	
			359	+ add \$t0, \$t0, \$t1	
			360	+ sw \$s2, (\$t0) # store the instruction cycle counter, icc	
			361	+ addi \$v0, \$s2, 0 # return the icc	
			362		

```

363 +      j      exit_start_cycle
364 +
365 + i2_onward:
366 +      beq     $a0, $a1, equal      # if a0 == a1
367 +      slt     $t0, $a0, $a1      # if a0 < a1, t0 = 1
368 +      bne     $t0, $0, goto_a1
369 +
370 + goto_a0:
371 +      la      $t0, DIFF           # DIFF holds the instruction start cycle of each
372 +      instruction
373 +      addi     $t1, $a0, -1        # get the address of DIFF[i]
374 +      sll     $t1, $t1, 2
375 +      add     $t0, $t0, $t1
376 +      lw      $a0, ($t0)          # t0 = DIFF[i]
377 +      jal     calc_diff
378 +      j      exit_start_cycle
379 +
380 + goto_a1:
381 +      la      $t0, DIFF           # DIFF holds the instruction start cycle of each
382 +      instruction
383 +      addi     $t1, $a1, -1        # get the address of DIFF[i]
384 +      sll     $t1, $t1, 2
385 +      add     $t0, $t0, $t1
386 +      lw      $a0, ($t0)          # t0 = DIFF[i]
387 +      jal     calc_diff
388 +      j      exit_start_cycle
389 +
390 + equal:
391 +      la      $t0, DIFF           # DIFF holds the instruction start cycle of each
392 +      instruction
393 +      addi     $t1, $a0, -1        # get the address of DIFF[i]
394 +      sll     $t1, $t1, 2
395 +      add     $t0, $t0, $t1
396 +      lw      $a0, ($t0)          # t0 = DIFF[i]
397 +      jal     calc_diff
398 +      j      exit_start_cycle
399 +
400 + exit_start_cycle:
401 +      addi     $s2, $s2, 1        # increment the icc
402 +      lw      $ra, 0($sp)
403 +      addi     $sp, $sp, 4
404 +      jr      $ra
405 +
406 + #####
407 + #
408 + #####
409 + .globl calc_diff
410 + calc_diff:
411 +      addi     $sp, $sp, -4        #save argument and $ra
412 +      sw      $ra, 0($sp)
413 +
414 +      sub      $t0, $s2, $a0      # t0 = current start cycle - recent start cycle
415 +      addi     $t1, $0, 3        # difference is at least 3 or greater
416 +      bge      $t0, $t1, exit_diff
417 +      addi     $t1, $t1, -1      # difference is 2
418 +      beq      $t0, $t1, diff_2  # difference is 1
419 +
420 + diff_1:
421 +      addi     $s2, $s2, 2
422 +      j      exit_diff
423 +
424 + diff_2:
425 +      addi     $s2, $s2, 1
426 +      j      exit_diff
427 +
428 + exit_diff:
429 +      addi     $v0, $s2, 0
430 +      lw      $ra, 0($sp)
431 +      addi     $sp, $sp, 4
432 +      jr      $ra
433 +
434 + #####
435 + # Gets the register associated with the instruction
436 + #####
437 + #####
438 + #####

```

```

435      addi     $sp, $sp, -4
436      sw      $ra, 0($sp)
437      addi     $t0, $0, 10        # i = size(dst) - 1
438 +      addi     $s7, $0, 0
439
440 compare_loop:      #for each element in DST
441      la      $t1, DEST
442      sll     $t2, $t0, 2
443
444
445
446
447
448
449
450
451
452
453 +      addi     $s7, $0, 1
454
455 exit_comp_loop:
456 +      addi     $t8, $t0, 1
457      lw      $ra, 0($sp)
458      addi     $sp, $sp, 4
459      jr      $ra
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

```

335 # Gets the register associated with the instruction
336 #####
337 #####
338 #####
339 #####
340
341 @@ -340,6 +435,7 @@ compare:
342      addi     $sp, $sp, -4
343      sw      $ra, 0($sp)
344      addi     $t0, $0, 10        # i = size(dst) - 1
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

381	jal	print_dependence	478	jal	print_dependence
382	exit_comp_loop2:		479	+ addi	\$s7, \$0, 1
			480	exit_comp_loop2:	
			481	+ beq	\$s7, \$0, print_none
			482	+ lw	\$ra, 0(\$sp)
			483	+ addi	\$sp, \$sp, 4
			484	+ jr	\$ra
			485	+ print_none:	
			486	+ la	\$4, MSG_NONE
			487	+ jal	print_string
			488	+ addi	\$t9, \$t0, 1
383	lw	\$ra, 0(\$sp)	489	lw	\$ra, 0(\$sp)
384	addi	\$sp, \$sp, 4	490	addi	\$sp, \$sp, 4
385	jr	\$ra	491	jr	\$ra

0 comments on commit 7c32b85

Lock conversation

Write

Preview

H B I ≡ <> 🔗 ≡ ≡ ≡ @ ↻ ↶

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.

Comment on this commit

Unsubscribe

You're receiving notifications because you're watching this repository.