

## 컴퓨터 비전 응용을 위한 VLIW 보조프로세서의 하드웨어 설계

최병윤\*

### Hardware Design of VLIW coprocessor for Computer Vision Application

Byeong-yoon Choi\*

Department of Computer Engineering, Dongeui University, Busan 614-714, Korea

#### 요 약

본 논문에서는 자동차용 컴퓨터 비전 알고리즘을 고속으로 처리하기 위해 VLIW 보조프로세서를 설계하였다. VLIW 보조프로세서는 8단 파이프라인 구조로 1개의 사이클에 4개의 명령을 처리할 수 있으며, 보행자 인식을 위한 36개의 정수 및 부동 소수점 명령어 집합을 갖고 있다. 프로세서는 45nm CMOS 공정에서 최대 동작 속도는 300-MHz이며 약 210,900 게이트로 구성되며 예상 연산 성능은 1.2 GOPS 이다. VPE와 8개의 VLIW 코어로 구성된 비전 프로세서 시스템은 25~29 FPS의 보행자 검출 성능을 가진다. VLIW 보조 프로세서는 높은 검출 속도와 호스트 프로세서와 느슨한 결합 특성으로 다양한 비전 분야에 응용 가능하다.

#### ABSTRACT

In this paper, a VLIW(Very Long Instruction Word) vision coprocessor which can efficiently accelerate computer vision algorithm for automotive is designed. The VLIW coprocessor executes four instructions per clock cycle via 8-stage pipelined structure and has 36 integer and floating-point instructions to accelerate computer vision algorithm for pedestrian detection. The processor has about 300-MHz operating frequency and about 210,900 gates under 45nm CMOS technology and its estimated performance is 1.2 GOPS(Giga Operations Per Second). The vision system composed of vision primitive engine and eight VLIW coprocessors can execute pedestrian detection at 25~29 frames per second(FPS). Because the VLIW coprocessor has high detection rate and loosely coupled interface with host processor, it can be efficiently applicable to a wide range of vision applications.

**키워드** : VLIW, 컴퓨터 비전, SoC, 자동운전보조시스템

Key word : vliw, computer vision, SoC, ADAS(Automatic Driving Assistance System)

접수일자 : 2014. 08. 04 심사완료일자 : 2014. 08. 28 게재확정일자 : 2014. 09. 05

\* **Corresponding Author** Byeong-Yoon Choi (E-mail:bychoi@deu.ac.kr, Tel:+82-51-890-1706)

Department of Computer Engineering, Dongeui University, Busan 614-714, Korea

**Open Access** <http://dx.doi.org/10.6109/jkiice.2014.18.9.2189>

print ISSN: 2234-4772 online ISSN: 2288-4165

©This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.  
Copyright © The Korea Institute of Information and Communication Engineering.

## I. 서 론

센서, 카메라 등을 통해 입력된 이미지를 처리하여 오브젝트 추출, 동작 인식을 수행하는 컴퓨터 비전은 최근에 자동차에 내장이 되는 자동운전보조시스템(ADAS, Automatic Driving Assistance System)과 스마트기기를 중심으로 시장이 급속하게 커지고 있다[1-4]. 두 분야의 컴퓨터 비전 응용은 다양한 조명과 기후 조건에서 높은 성능과 낮은 클럭 주파수와 저전력 특성을 요구한다. 자동차 용 비전 시스템은 경계선, 차선과 같은 길의 구조와 함께 자동차, 보행자와 같은 물체를 고속으로 검출하는 능력을 요구한다. 컴퓨터 비전 기술이 고급화됨에 따라 카메라와 센서 해상도가 커지고 요구 인식이 높아지고, 동시에 저전력을 요구함에 따라 기존 소프트웨어와 범용 하드웨어로는 이를 만족하기 어려워, 최근 특화된 구조의 비전 프로세서에 대한 연구가 활발하다. 특히 ADAS 분야에서는 Mobileye사의 EyeQ 시리즈가 전 세계 시장의 60%를 점유함으로써 비전 프로세서의 가능성을 보여준 바 있다[5]. EyeQ사의 비전 프로세서는 고성능 RISC 마이크로프로세서와 다수개의 특화된 비전 코어들로 구성되어 병렬 처리를 할 수 있는 구조이다.

본 논문에서 제안하는 비전 시스템은 알고리즘 변화에 대처할 수 있도록 프로그램 가능한 구조의 보조프로세서, 고정 기능의 고성능의 전용 비전 하드웨어와 호스트 프로세서가 결합한 구조를 갖고 있다. 특히 다양한 분야의 비전 알고리즘을 가속화할 수 있도록, 높은 연산 능력과 넓은 대역폭을 갖고 있어 멀티미디어, 그래픽스와 비전 알고리즘에 적합한 연산 구조로 평가되는 VLIW(Very Long Instruction Word) 구조를 보조프로세서로 사용하였다[6-8]. 본 논문에서는 자동차용 비전 시스템에 내장되어 다양한 비전 알고리즘을 고속으로 구현할 수 있는 VLIW 비전 코어를 설계하고, 비전 시스템에 적용하여 동작을 확인하고 성능을 분석하였다.

본 논문의 구성은 다음과 같다. 2 장에서는 VLIW 프로세서의 구조를 기술하고, 3 장에서는 VLIW 보조프로세서의 설계 사양과 하드웨어 설계를 기술하였으며, 4 장에서는 설계한 회로에 대한 검증과 성능 분석을 하였으며 마지막으로 결론을 기술하였다.

## II. 비전 프로세서(VP)의 구조

비전 시스템을 구현하는 비전 프로세서(VP)는 VPE (Vision Primitive Engine), VCU (Vision Core Unit), 제어 회로, 호스트 프로세서로 구성된다. 본 연구에서 설계하는 비전 코어(VC)는 VLIW 구조를 갖고 있으며, 응용 분야의 성능 요구에 따라 가변적인 수가 VCU 내에 내장되어 병렬 처리된다. 그림 1은 비전 프로세서(VP)의 전체 블록 다이어그램이다. 비전 코어(VC)는 음영 처리되어 표시되며, 현재 구현한 VP에서는 8개의 VC를 내장하고 있다.

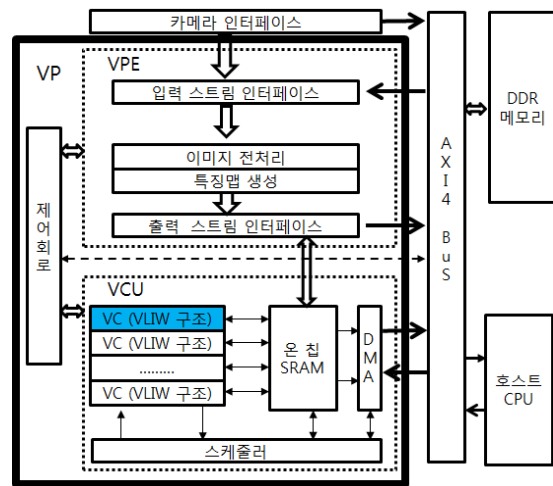


그림 1. 비전 프로세서(VP)의 블록도  
Fig. 1 Block diagram of vision processor(VP)

VPE는 영상 전처리 작업과 특징 지도(Feature Map)를 고속으로 생성하고 이를 DDR 메모리에 저장한다. VPE가 수행하는 동작은 부동 소수점 데이터에 대한 연산으로 구성되고, 빠른 속도를 위해 전용 하드웨어 구조로 구성하였다. 비전 코어 유닛(VCU)은 온칩 SRAM, VC, 스케줄러와 DMA로 구성된다. 다양한 크기의 윈도우에 대해 물체 인식 알고리즘을 고속으로 처리하기 위해 SRAM은 특징맵과 메타 데이터(Meta Data)가 사전에 적재되며, 다수 개의 VC가 이를 이용해 물체 인식 비전 알고리즘을 병렬로 처리한다. 다수 개의 비전 코어(VC)가 동시에 SRAM 메모리에 접근할 수 있으므로 메모리 충돌을 최소화하기 위해 크로스바 스위치 연결 구조로 연결된다. 스케줄러는 VCU 상태와 제어를

관장하는 상태 머신으로 슬라이딩 윈도우 방식으로 VC 태스크 할당, 스케줄링과 동기화 작업을 수행한다. 제어회로는 명령 큐 구조를 갖고 있으며, 호스트 CPU로부터 일련의 명령을 받아서 각 세부 블록에 제어 신호를 전달하는 기능과 비전 프로세서 내의 상태 정보를 주기적으로 검사하는 기능을 수행한다.

### III. VLIW 구조의 비전 코어(VC) 설계

VC는 VCU 내에 8개가 있으며, 스케줄러에 의해 활성화가 제어되는 보조프로세서로 동작한다.

#### 3.1. 설계 사양

VLIW 구조의 비전 코어(VC) 보조프로세서는 다음과 같은 설계 사양을 갖는다.

첫째, 자동차용 보행자 인식 알고리즘에서 물체 인식 알고리즘을 가속화하도록 한다.

둘째, VLIW 구조의 비전 코어와 호스트 프로세서 간에 동작의 융통성을 극대화하기 위해 느슨하게 결합되어, 비전 프로세서는 이종(heterogeneous) 프로세서 구조를 갖는다.

셋째, 높은 인식률과 정밀도를 위해 기존 DSP용 VLIW 프로세서[5]와 달리 고정 소수점 데이터가 아닌 단정도 부동소수점 데이터 형식을 지원한다.

넷째, 연산 시 오버플로우와 언더플로우 발생 시 중단이 없는 실시간 처리를 위해 포화 연산 방식을 지원한다.

다섯째, 멀티미디어와 비전에서 요구하는 데이터 특성에 따라 명령에 사용되는 오퍼랜드는 32-비트, 64-비트 정수와 함께 2개의 32-비트의 벡터 데이터의 3가지 형태를 지원한다.

여섯째, 128-비트의 패킷 명령어는 항상 4개의 명령을 담고 있는 구조를 갖도록 하여, 명령어 인출 회로와 명령어 캐쉬 메모리 제어를 단순화하였다.

일곱째, 자동차 분야에 비전 응용을 고려한 특화된 명령어 집합과 주소 지정모드를 정의한다.

여덟째, 필요한 레지스터 공간을 확대하기 위해 이중뱅크 구조의 레지스터 파일을 사용한다.

아홉째, 참고 문헌 [3]에서 정의한 자동차용 비전 시스템의 15~30 FPS 검출 속도를 성능을 목표로 한다.

#### 3.2. 명령어 형식

VLIW 프로세서는 <그림 2>와 같이 1개의 패킷명령에 4개의 세부 명령을 포함하는 형태로 구성된다. 각 명령은 기존 RISC 프로세서의 명령어와 유사한 형식과 기능을 가지고 있다. 단, 기존 RISC와 달리 32-비트 상수 값을 지원하기 위해, 32-비트 즉시 데이터(immediate data)를 내장한 패킷 형식 1의 64-비트 명령어 형식을 갖는다. 32-비트보다 작은 즉시 값을 2개 결합하여 생성하는 방안을 검토했지만, 8단계의 파이프라인 단계의 데이터 의존 특성으로 요구하는 성능을 맞추기 어려워, 32-비트 즉시 데이터를 지원하는 방안을 사용하였다.

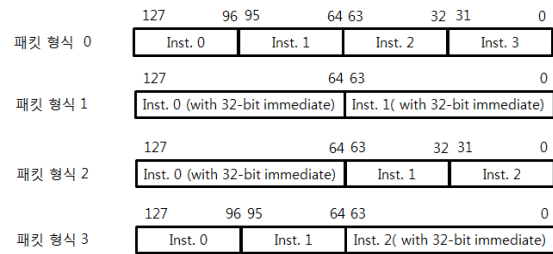


그림 2. 명령어 패킷의 형식  
Fig. 2 Format of instruction packet

명령어 패킷에 포함되는 세부 명령어는 크게 산술 명령, 부동소수점 연산 명령, 비교 동작을 하는 명령, 분기 명령 및 시스템 제어 명령으로 구분된다. 단, 보조프로세서로 동작하여 함수 호출 및 복귀, 인터럽트 명령은 갖고 있지 않다. <표 1>는 VLIW 프로세서에서 사용하는 주요 명령을 나타낸다. 비전 알고리즘은 반복적인 병렬 연산 동작과 윈도우 형태의 복잡한 데이터 접근 형태, 정밀도를 위한 부동 소수점 연산이 필요하다. 이러한 동작 특성을 반영하여, 특화된 명령어를 정의하고 구현하였다. 반복 동작을 분기 명령으로 구현할 경우, 종료 조건 판단 과정에 제어 의존성에 따른 코드 낭비가 존재하므로, 지연이 없는 하드웨어 루프를 구현하는 명령(LOOP#, ENDLOOP#)을 갖고 있다. 현재 2개의 하드웨어 루프를 지원하고 있으며, 기존 RISC에서 분기 명령 수를 최소화하기 위해 비교 명령의 경우 비교 결과를 조건 코드(condition code)가 아닌 predicate 레지스터에 참/거짓 형태로 저장하여, 일부 명령은 그 값으로 조건적인 수행을 할 수 있도록 하였다.

표 1. 주요 명령어

Table. 1 Subset of instruction set

유형	어셈블리 형식	
Load	LDD RD, [RS, #S11]	
	LDD DD!, [RS, #S11]	
	LDW RD!, [DS, #U4, #u4]	
	LDW XD, [RS, #S11]	
Store	STD RD, [RS, #S11]	
	STD DD!, [RS, #S11]	
Control Transfer	B #DISP	
	J RS	
	LOOP# #U13	
	ENDLOOP#	
Integer Arithmetic (ALU)	MOV RD, CS	
	MOV CD, RS	
	MOV RD #U32	
	MOV RD, #S16	
	CONCAT DD, RS, RT	
	ADD RD, RS, RT	
	ADD RD, RS, #s16	
	NOP	
	TABLEINDEX RD, RS, #U4, #U5	
	KEYGEN0 RD, PS	
	KEYGEN2 RD, PS	
	KEYGEN3 RD, PS	
	Floating-point	FADD RD, RS, RT
		FSUB RD, RT, RS
FMUL RD, RS, RT		
FCMPLT PD, RS, RT		
F2U RD, RS		
Floating-point vector	VFCMPLT PD, DS, DT	
	VFSUB DD, DT, DS	
	VFSUBC DD, DS, DT	
System control	REGSWITCH	
	(PU) ENDP #u8	

그리고 32비트 데이터와 함께 64비트 오퍼랜드를 처리하는 명령어 함께 존재하는데, 64-비트 오퍼랜드의 경우 시작 레지스터 주소는 짝수 값을 갖도록 제한된다. 룩업 테이블(lookup table)에 대한 접근을 가속화하기 위한 TableIndexw 명령, predicate 레지스터를 사용한 메타 데이터에 대한 키 상수를 만드는 KegGen 명령 등의 특화된 명령어가 존재한다. Floating-Point Vector 덧셈과 뺄셈, 비교 동작은 64-비트 데이터를 받아 32-비트 단위로 2개의 벡터 부동 소수점 연산을 지원한다. 단, 부동 소수점 곱셈 명령은 포화 연산을 지원하는 1개의 32-비트 곱셈 동작을 수행하며, 비교 동작은 정수와 부동 소수점 데이터에 대해 32-비트 비교와 64-비트 벡터

비교를 모두 지원한다. 프로그램 흐름을 바꾸는 제어 전달(control transfer) 명령은 PC-relative, register-ndirect jump를 지원한다. 또한 VLIW 프로세서가 보조 프로세서로 동작하므로, 프로그램 종료를 외부 스케줄러에게 알리는 명령이 존재한다. 메모리 참조하는 명령은 기존 load/store 명령과 유사하지만, 8개의 VLIW 프로세서가 하나의 공유 메모리를 공유해서 사용하므로 외부 스케줄러에게 특정 메모리 번지를 요청하고, 명령의 수행을 대기하는 기능이 존재한다. <그림 3>은 명령어 패킷에 속한 산술 명령의 형식을 나타내며, 명령어의 C 비트는 패킷내 마지막 명령의 경우 “1”을 가진다.

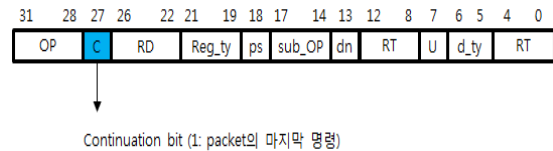


그림 3. ALU 명령 형식  
Fig. 3 Format of ALU\_ instruction

3.3. 명령어 파이프라인 구조

VLIW의 모든 명령 패킷은 순서대로 인출 및 완료하는 구조로 <그림 4>와 같은 명령어 패킷 파이프라인 구조를 갖는다. IF(instruction fetch) 단계는 명령어 캐쉬 메모리에서 128-비트 명령어 패킷을 인출하고, DP(dispatch) 단계는 명령어 패킷을 4개의 슬롯에 맞게 명령을 정렬해서 전달하는 역할을 하며, ID/OF 단계에서 필요한 디코딩 신호를 생성한다. 실행단계는 4개의 단계 EX1~EX4로 구성된다. 단, 정수 데이터와 부동 소수점 데이터는 EX1 ~ EX3 단계에서 동작을 완료하지만, 순서대로 종료하는 방식에 따라 1, 2단계의 지연을 통해 레지스터 파일에 저장된다. 반면 메모리 참조 명령은 주소 생성은 EX1단계에서 이루어지고, 공유 메모리의 자원 충돌, 중재 등의 동작 처리로 EX2 ~EX4(MEM1~MEM3)의 3단계 파이프라인으로 구현된다. 마지막으로 WB(write-back) 단계는 레지스터 파일(register file)에 쓰기 동작을 수행한다. 분기 명령은 PC-relative 주소 방식의 경우 ID단계에서 분기가 이루어지고, 나머지 분기 동작은 EX1단계에서 이루어진다.

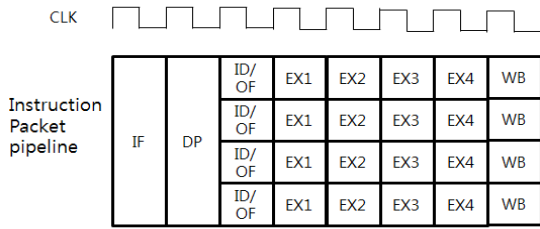


그림 4. 명령어 패킷 파이프라인  
Fig. 4 Pipeline of instruction packet

3.4. 명령어 슬롯별 처리가능 명령

<표 2>는 명령어 슬롯별로 처리 가능 명령어 유형을 나타낸다.

표 2. 명령어 슬롯별 명령 유형  
Table. 2 instruction types per slot

슬롯	수행 가능 명령 유형
슬롯 0	ALU, MOV immediate 명령(64-bit), compare 명령, memory access 명령
슬롯 1	ALU 명령, FP 명령, compare 명령
슬롯 2	ALU 명령, FP 명령, compare 명령, MOV immediate 명령(64-bit)
슬롯 3	ALU 명령, Control Transfer 명령, System Control 명령, compare 명령

3.5. VLIW 비전 코어의 하드웨어 설계

VLIW 비전 코어 프로세서는 <그림 5>와 같은 구조를 가지고 있다. 명령어 캐시 메모리는 직접 사상 구조를 가지고 있으며, 하나의 블록에는 4개의 명령어 패킷을 저장하는 구조로 1-k 패킷 워드를 저장한다. 비교 기능을 내장한 정수 처리 장치(IU)는 모두 64-비트 구조이며, 슬롯 0과 1의 부동 소수점 처리 장치(FPU)는 32-비트 구조이지만 벡터 처리를 위해 두 개의 쌍으로 구성된다. 슬롯 0에는 메모리 참조를 위한 하드웨어가 존재한다. 슬롯 3에는 분기 동작과 하드웨어 루프 처리 및 시스템 제어를 위한 하드웨어를 갖고 있다. 레지스터 파일은 범용 레지스터 파일(GPR)과 제어용 레지스터가 존재한다. 단, 범용 레지스터 파일은 64-비트와 32-비트 단위로 접근이 지원되며, 명령어의 레지스터 필드의 길이를 변화시키지 않고, 레지스터 수를 증가시키기 위해 <그림 6>과 같이 전방 뱅크 0(bank 0)과 후방 뱅크(bank 1)의 이중 뱅크 구조로 구성하였다.

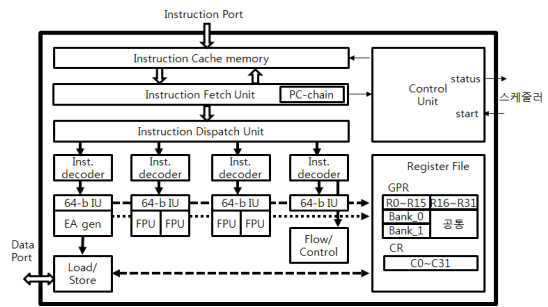


그림 5. VLIW 구조 비전 코어의 구조  
Fig. 5 Architecture of VLIW Vision core

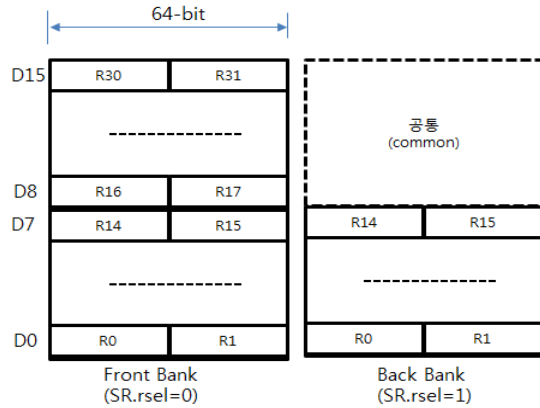


그림 6. 뱅크 구조로 된 범용 레지스터 파일  
Fig. 6 register file with bank structure

뱅크 0는 뱅크 1과 R16 ~ R31을 공유하므로 실제 물리적인 레지스터 개수는 48개이다. 레지스터 파일은 제어 레지스터 SR(C2)의 rsel 비트에 따라 명령어가 접근하는 뱅크가 구분된다. 단, <표 1>의 Load와 같은 메모리 참조 명령의 경우 RD 혹은 DD 필드에 {!}를 지정하여, 뱅크 1에 결과를 강제 할당할 수 있도록 한다. 기본적으로 레지스터 파일은 64-비트 단위로 접근되며, 32-비트 데이터가 필요한 경우 읽은 64-비트 데이터 중 필요한 32-비트 오퍼랜드를 선택해서, 상하 32-비트를 동일한 값으로 만들어 제공한다. 제어 레지스터는 상태 레지스터(SR), 윈도우 위치 레지스터(WPOS), Predicate 레지스터, 루프 카운터, 메모리 동기를 위한 X0, X1 레지스터, 임시 레지스터(T0~T7) 등이 존재한다. <그림 7>은 부동소수점 데이터를 양의 정수로 변환하는 F2U 명령 하드웨어 구조를 나타낸다.



표 3. 성능 비교

Table. 3 Performance comparisons

	DSP 코어[7] (8개)	본 논문의 VLIW 보조프로세서 (8개)
구조 특성	FPU 명령 없음	S/W 파이프라인 적용
FPS@300MHz	1.3~1.5	25~29
cycle per loop	600	30

<그림 8>은 8개의 VLIW 프로세서가 내장된 비전 시스템의 FPGA 보행자 인식 검증 환경을 나타낸다. 카메라에서 영상을 받아서 USB 포트로 DDR 메모리에 프레임 전송한 후, 비전 프로세서가 보행자 검출 알고리즘을 수행한 후, 검출 결과를 PC에 전달하며 이전에 저장한 영상과 검출 결과를 이용해서 보행자 인식 결과를 화면에 박스로 표시한다. FPGA 보드 환경에서는 52 MHz에서 4~5 FPS의 동작 성능을 보였다. 구현한 비전 시스템은 높은 검출 속도와 호스트 프로세서와 느슨한 결합 특성으로, 내장되는 VLIW 보조프로세서 수를 조절하면 다양한 성능 요구조건을 갖는 비전 분야에 효율적으로 응용 가능하다.

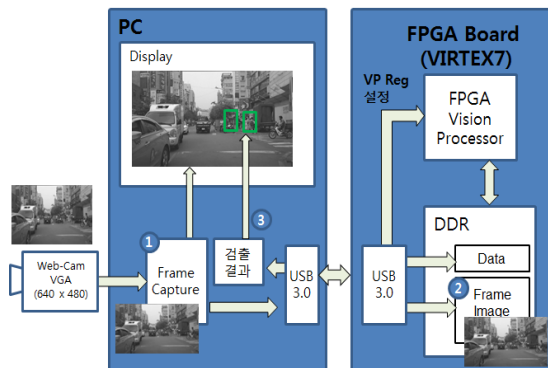


그림 8. 비전 프로세서의 검증 환경  
Fig. 8 Verification System for Vision processor

설계한 VLIW 구조의 비전 보조 프로세서를 특성은 <표 4>와 같다. VLIW가 높은 성능을 얻을 수 있는 이유는 부동 소수점 데이터를 지원하고 비전 응용에 최적화된 명령어 집합을 사용함에 기인한다.

표 4. VLIW 비전 보조프로세서의 특성

Table. 4 Characteristics of VLIW vision coprocessor

구조	4-way VLIW 구조
파이프라인 단계 수	8
패킷 길이	128-비트
패킷당 명령어 수	4 instructions / packet
지원하는 명령어 수	36
보행자 인식을 위한 특화 명령어 수	13 (Tableindex, keygen 등)
부동 소수점 처리 명령 유형	2-way vector FP 연산과 1-way FP 연산
게이트 수	210,900 @45nm CMOS 공정
동작 주파수	300 MHz @45nm CMOS 공정
동작 주파수(FPGA)	52 MHz@Virtex 7 (XC7A200T)
8개의 VLIW를 내장한 경우 FPS	25~29@300MHz, 8 VLIW

## V. 결론

설계한 VLIW 보조프로세서는 보행자 인식을 위한 비전 시스템에 내장되어 프로그램 가능한 보조프로세서로 동작하며, 패킷당 4개의 명령어를 처리하여, 특화된 명령어 집합과 윈도우 데이터를 위한 주소 지정 모드를 갖고 있다. 45nm CMOS 표준 셀 공정 조건에서 약 210,900 개의 게이트로 구성되며, 300 MHz의 동작 주파수로 최대 1.2 GOPS의 연산 성능을 가지고 있다. 자동차 보행자 인식을 위한 비전 시스템에 8개의 VLIW 보조 프로세서를 내장할 경우 300 MHz의 동작 주파수로 약 25~29 FPS의 성능을 갖고 있어서, 자동차 분야에서 요구하는 15~30의 FPS 조건을 만족한다. 적용되는 비전 응용 분야의 성능 요구 사항에 따라 비전 시스템에 내장되는 VLIW 보조 프로세서 수를 조정하면, 다양한 비전 응용에 내장될 수 있어서 설계된 VLIW 보조 프로세서는 하드웨어 IP(Intellectual Property)로 활용이 가능하며, VLIW 보조프로세서에 내장되는 명령어 집합을 추가하면 음성 인식 등 다른 분야에도 응용이 가능할 것으로 판단된다.

### 감사의 글

본 연구는 2013년도 동의대학교 연구년 연구비 지원에 의해 이루어진 연구로서, 회로 설계에 IDEC 지원 CAD 소프트웨어가 사용되었으며, 비전용 VLIW 프로세서의 설계 사양 결정에 (주)넥서스칩스의 자문을 받았습니다.

### REFERENCES

- [ 1 ] Gideon P. Stein, Elchanan Rushineek, Gaby Hayun, and Amnon Shashua, "A Computer Vision System on a Chip : a case study from the automotive domain", *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) Workshops*, Volume 03, pp. 130-134. 2005.
- [ 2 ] Jeff Bier, "Computer Vision in Mobile Devices : The Next Killer App ?", Linley Tech Mobile Device Conference, 2011.
- [ 3 ] S.Kyo and S. Okazaki, "IMAPCAR: A 100 GOPS in-vehicle vision based 128 ring connected VLIW processing elements," *Journal of Signal Processing Systems*, vol.62, pp.5-16, 2011.
- [ 4 ] Winfried Gerke, Jorn Jachalsky, Martin Wahle, "Flexible Coprocessor Architectures for Ambient Intelligent Applications in Mobile Communication and Automotive Domain," *Proc. of SPIE*, vol.5117, pp.310-320, 2003.
- [ 5 ] MobilEye Inc. EyeQ2TM Vision System on a Chip [Internet]. Available: <http://www.mobileye.com>.
- [ 6 ] Joseph A Fisher, Paolo Faraboschi, and Cliff Young, *Embedded Computing: A VLIW Approach to Architecture, Compilers, and Tools*, MK, 2005.
- [ 7 ] QualComm, "Hexagon V2 Programmer's Reference Manual", 80-NB419-1 Rev.A, August 4, 2011.
- [ 8 ] Chang-Hyo Yu, Kyusik Chung, Donghyun Kim, Lee-Sup Kim, "An Energy-Efficient Mobile Vertex Processor With Multithread Expanded VLIW Architecture and Vertex Caches", *IEEE Journal of Solid State Circuits*, vol. 42, no. 10, pp.2256-2269. Oct., 2007.
- [ 9 ] Mark Zwolinski, *Digital System Design with SystemVerilog*, Prentice-Hall, 2009.
- [ 10 ] M. Lam, "Software pipelining: An effective scheduling technique for VLIW machines", *PLDI 88*, pp. 318-328, July 1988.



최병윤(Byeong-Yoon Choi)

1985년 2월 : 연세대학교 전자공학과 졸업  
1987년 2월 : 연세대학교 공학석사  
1992년 8월 : 연세대학교 공학 박사  
1993년 3월 ~ 현재 : 동의대학교 컴퓨터공학과 교수  
※ 관심분야 : VLIW 프로세서 설계, RISC 설계, 네트워크 프로세서 설계, 정보통신 알고리즘의 SoC 설계